
摘要

城市公交线路仿真系统是应用科技手段提升公交管理水平和调度技能，如开发动态线路仿真系统，运用当代通讯技术研发科技公交。随着社会经济的高速发展以及城镇化进程的加快，人们对于公共交通的质量与效率的追求不断提高，公共交通需求日益多样化，其中城市公交作为最初始的公共交通尤为重要。由此可见，城市公交线路仿真系统能在一定程度上压力帮助城市缓解公共交通压力。

本课题旨在基于 Java EE 的基础上，开发一款城市公交线路仿真系统。运用前后端分离技术，通过使用 VUE + Springboot 的逻辑结构开发，使用 VUE 完成前端的开发，并结合 Springboot 的后端开发技术，利用跨域连接前后端，完成系统开发。

本文主要从选题背景与意义、可行性与需求分析、功能设计与实现、测试等四个方面，按照制作的流程对本次设计进行论述，对开发的思路、实现以及实现过程中的主要问题都予以描述记录。

关键词：城市公交线路；深圳公交；VUE；Springboot；前后端交互

Simulation system of urban public transportation line based on Java

EE

Abstract

Urban bus line simulation system is the application of scientific and technological means to improve the level of bus management and scheduling skills, such as the development of dynamic line simulation system, the use of contemporary communication technology to develop scientific and technological buses. With the rapid development of the social economy and the acceleration of the urbanization process, people's pursuit of the quality and efficiency of public transportation continues to increase, and public transportation needs are increasingly diversified. Among them, urban public transportation is particularly important as the initial public transportation. It can be seen that the urban bus line simulation system can help the city relieve the pressure of public transportation to a certain extent.

This subject aims to develop a city bus line simulation system based on Java EE. Using front-end and back-end separation technology, through the use of VUE + Springboot logical structure development, using VUE to complete the front-end development, combined with Springboot's back-end development technology, the use of cross-domain connection front-end and back-end, complete system development.

This article mainly discusses the design according to the production process from four aspects: the background and significance of the topic selection, the feasibility and demand analysis, the functional design and function implementation, and the testing. The problem is described and recorded

Keywords: Urban bus lines; Shenzhen public transport; VUE; Springboot; Front and back interaction

目 录

1. 前言.....	1
1.1. 选题背景.....	1
1.2. 本系统在国内外的的发展概况及存在的问题.....	1
1.3. 本系统的目的、意义及应达到的技术要求.....	2

1.4. 本系统应解决的主要问题.....	2
2. 可行性研究.....	3
2.1. 技术可行性.....	3
2.2. 经济可行性.....	3
2.3. 总结.....	3
3. 概要设计.....	4
3.1. 系统框架设计.....	4
3.2. 系统功能模块设计.....	6
4. 功能详细设计与实现.....	10
4.1. 线路查询功能设计与实现.....	10
4.2. 线路创建设计.....	11
4.3. 线路编辑设计.....	13
4.4. 线路汇总设计.....	14
4.5. 线路仿真设计.....	15
4.6. 登录与注册功能设计.....	16
4.7. 权限控制设计.....	19
4.8. 数据设计.....	20
5. 测试.....	21
5.1. 测试思路与方法.....	21
5.2. 基础测试.....	22
5.3. 具体功能测试.....	23
5.4. 测试总结.....	24
6. 总结.....	25
参考文献.....	26
谢 辞.....	27
附 录.....	28

1. 前言

1.1. 选题背景

随着社会经济的高速发展以及城镇化进程的加快，人们对于公共交通的质量与效率的追求不断提高，公共交通需求日益多样化，其中城市公交作为最初始的公共交通尤为重要。根据城市公交现状问题分析与线网布局规划模式研究^[10]，城市公交系统一般包含公交线路、站点、运营支持系统四个方面，它们相互配合共同作用。

近些年来，城市公交发展问题一直都是每一个城市都需要解决的重要问题，许多问题都需要通过系统分析并解决，如公交线路覆盖不全面，公交运营时间不当等问题，城市公交问题已经成为制约社会可持续发展的首要问题。优化公交运营系统性能已然成为解决城市发展问题和提高市民生活质量的首要目标。通过设计城市公交线路仿真系统，我们就可以使得公交线路管理更加的规范与高效，所以笔者选择设计基于 java EE 的城市公交线路仿真系统优化与解决城市公交治理问题和优化城市公交管理水平

1.2. 本系统在国内外的的发展概况及存在的问题

根据浅谈国内外交通仿真发展概况，不难发现城市交通仿真技术成长最快和最先的国家就是美国，世界上其他国家的仿真软件都是在美国的交通仿真技术的基础上进行进一步的发展，美国早在 1967 年就有计算机专业的专家就组织成立了美国的计算机仿真学会 (Society for Computer Simulation)，仿真学会的成立极大的推进了美国在城市交通仿真方面的技术钻研的发展^[22]。世界上很多国家才逐渐开始对交通仿真技术的感兴趣且开始研究，期间也源源不断有新的仿真软件出现，可是大部分仿真软件都不能实际投入生产使用中。如今已经到了 21 世纪，已经有大部分交通仿真软件都已经实现了交通仿真软件的商业化，并且投入实际生产使用中，就如西班牙 TSS 公司开发的 AIMSUN2，是一款交互式交通仿真软件，适用于检验与测试交通控制系统，它还能够用于对交通状况的进行预测以及优化车辆导航系统和其他实时交通信息的应用。

与国外相比，在那个年代国内整个交通行业发展薄弱，更别说道路交通系统仿真方面研究有什么起色。早在 20 世纪 80 年代，我国交通仿真技术发展初期，大部分技术研究都是集中在高等院校等研究机构进行研究，主要是为了解决一些相对简单的交通问题进行研究模拟，可是这些研究对于交通实际问题的解决并没有起到太大的作用。经过许多年的研究与摸索后，我国交通工程的相关研究人员才开始意识到需将研究方向改为如何实际性的解决交通问题，此后就开始钻研相关的建设性研究。如在 2014 年东北大学徐晓昱设计的基于 Flexsim 的城市公交系统仿真，用数学模型来描述公交系统微观层次

的细节存在局限性，离散虚拟现实的仿真技术作为一种重要的研究手段可以有效避免这一缺陷^[20]。

2020 年国内交通仿真软件的研发已经处于一个高速发展的阶段，从技术层面上看，我国的交通仿真技术已经属于世界顶级水平，虽然当初我国基础技术和人才方面都较为单薄，但是通过我国的科技人才不断努力的研发，如今已经能达到商用化水平，不过还是有许多的问题需要优化与解决，例如交通线路规划管理、交通运营方案，仿真软件设计，公交线路数据采集等许多技术上的问题。

1.3. 本系统的目的、意义及应达到的技术要求

本次毕业设计主要集中研究城市公交线路规划类型，并设计制作一款城市公交线路仿真系统，我们就可以做到公交信息管理的系统化、规范化和自动化。将传统公共交通系统的开环方式不断优化，调整为闭环反馈的整体调控过程，且更加高效，从而建立体系辨识、状态与演变再现、趋向善于分析和预测的分析能力，实现一种全新概念的智能管理系统和手段，从而优化城市公交的治理问题。

城市公交作为突显城市文明的主要窗口行业之一，不仅展示着一个城市的文明程度，还反映了城市科技发展水平。改革开放以来，深圳市公共交通事业发展迅速，在推进城市建设的进程中发挥了重要作用，同时也说明了深圳市科技水平发展迅速。

通过本系统，有助于将城市公共交通建设与管理水平提升到一个新的高度。基于 Java EE 的基础上，进行 web 设计，通过 Js, html5, css, 等编程技能实现网页系统，公交管理人员可以利用本系统进行线路查询及管理，有效地调度公交系统。

1.4. 本系统应解决的主要问题

由深圳市公共交通总体规划^[12]以及深圳市城市交通仿真系统，从中分析得到的现阶段调度水平落后是目前公交营运效率低下的主要原因。主要是调度方法单一，缺少灵活性，较少使用先进科技技术手段研制仿真系统，缺乏应变能力。可以通过科技手段提升公交调度与管理水平，如开发路线仿真系统，方便管理人员运用现代通讯技术调剂公交运营，高效地进行公交调度及管理。目前由于许多城市的道路规划出现问题，导致城市交通拥挤状况加剧的例子源源不断，本系统能通过模拟线路规划，收集数据，从而优化道路规划引起的城市交通拥堵问题。

2. 可行性研究

2.1. 技术可行性

本次设计是基于 Java EE 基础上，使用 JavaScript、Html5、CSS 等编程语言进行前端开发的。Vue.js 是当前较为常用的前端框架之一，它帮助开发人员快速地构建与开发前端项目，是一套方便开发人员构建用户界面的渐进式框架，Vue 的核心库主要专注于视图层，并且很方便的与其他第三方库或现有项目进行集成。在当下比较流行的两种 Java EE 框架有 SSH (Struts、Spring、Hibernate)，SSM (Spring、SpringMVC、MyBatis)。

SSH 一般指的是 Struts2 做控制器(controller)，spring 管理各层的组件，hibernate 负责持久化层，SSH 较重视配置开发，此中的 Hibernate 对 JDBC 的完整封装更面向对象，对增删改查的数据维护更自动化，但 SQL 优化方面较弱，且入门门槛稍高。并不十分适合学生个人开发者。

SSM 则指的是 SpringMVC 做控制器(controller)，Spring 管理各层的组件，MyBatis 负责持久化层。SSM 偏向轻量级配置，将注解开发发挥到极致，且 ORM 实现更加灵活，SQL 优化更轻便，较为适合像笔者这样的学生个人开发者。

由上可知，SSM 框架为初级的开发者给予了很大的便利，且能更好的掌控 SQL，这会让有数据库经验的个人开发者能开发出更高效率的 SQL 语句，并且 XML 配置管理起来也非常方便。综上所述，本次设计在基于 Java EE 基础上利用前后端分离方法，使用 SSM 框架进行后端开发，使用 Vue 进行前端开发，在技术上可行。

2.2. 经济可行性

本次毕业设计为城市公交线路仿真系统，在 Java EE 基础上进行网页开发，只需通过网页访问，就能进行公交线路管理仿真。由于本次系统为笔者独立开发，故不需考虑团队开销的问题，本系统的成本主要集中在前期资源的搜集与系统开发上，所以本系统投入成本少，若作为实际系统投入使用，对于城市公交管理人员，本系统还是有一定的吸引力，所产生的效益会远大于系统开发成本。综上所述，本次毕业设计在经济上可行。

2.3. 总结

从技术层面上来说，前后端分离开发，使用 Vue + SSM 框架十分的适合本次论文设计；从经济层面来说，本次论文设计由于设计规模较小，投入的开发成本较少。综合以上论述表明，本次设计在基于 Java EE 的基础上，使用 Vue + SSM 框架开发一套城市公交线路仿真系统，是可行的。

3. 概要设计

3.1. 系统框架设计

通过对比类似的管理系统的学习研究,为了本论文题目设计了与之合适的基础框架,本次系统使用前后端分离技术,将系统分为前端部分、后端部分、数据库部分。前端部分主要是 vue 前端框架(基于 webpack 构建)如图 4.1,Vue 中有许多的组件如 vuex, vue-router, vue-resource,再加上构建工具 vue-cli,sass 样式,就是一个完整的 vue 项目的主要组成。后端部分为 Spring+SpringMVC+Mybatis 后台开发框架(基于 maven 构建)如图 4.2。各部分间的关系见下图 4.2。

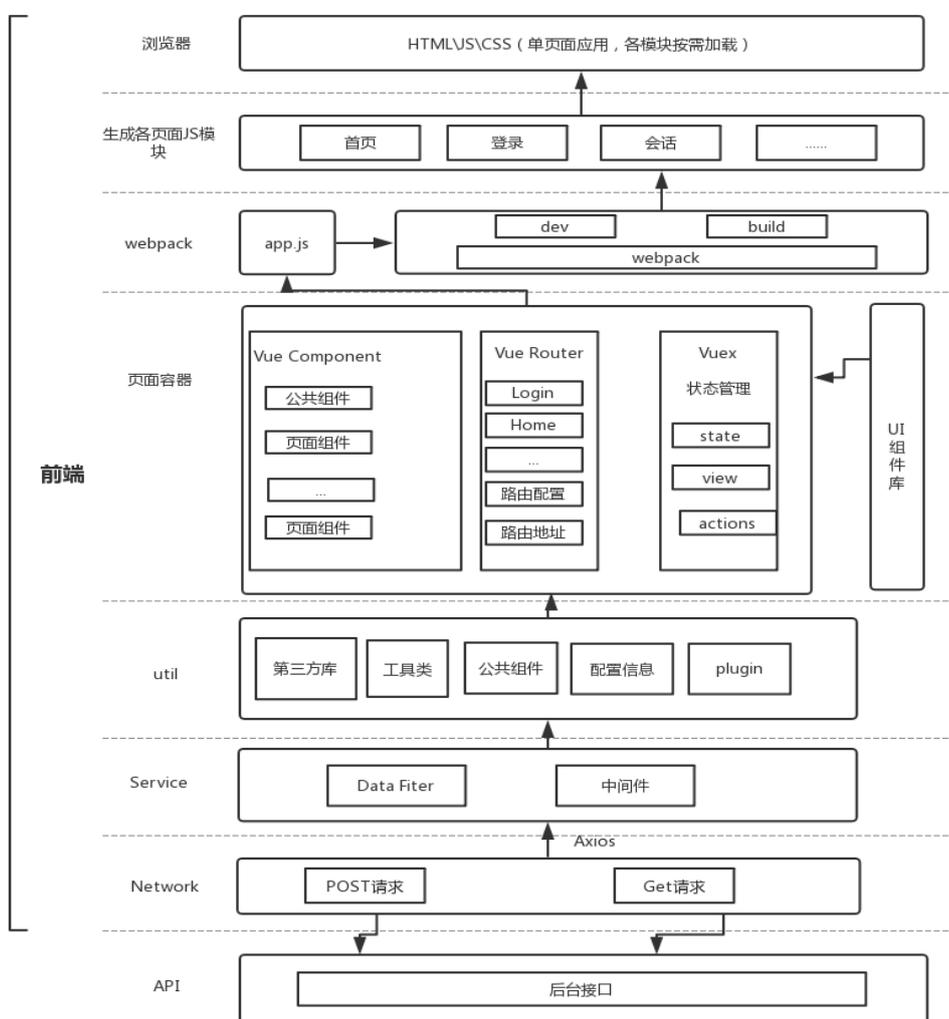


图 4.1 Vue 架构图

程序执行流程图

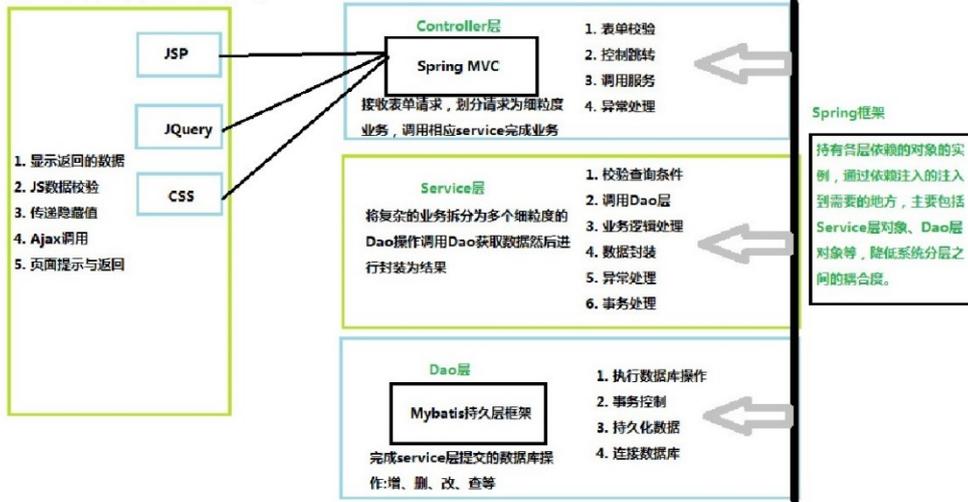


图 4.2 SSM 框架图

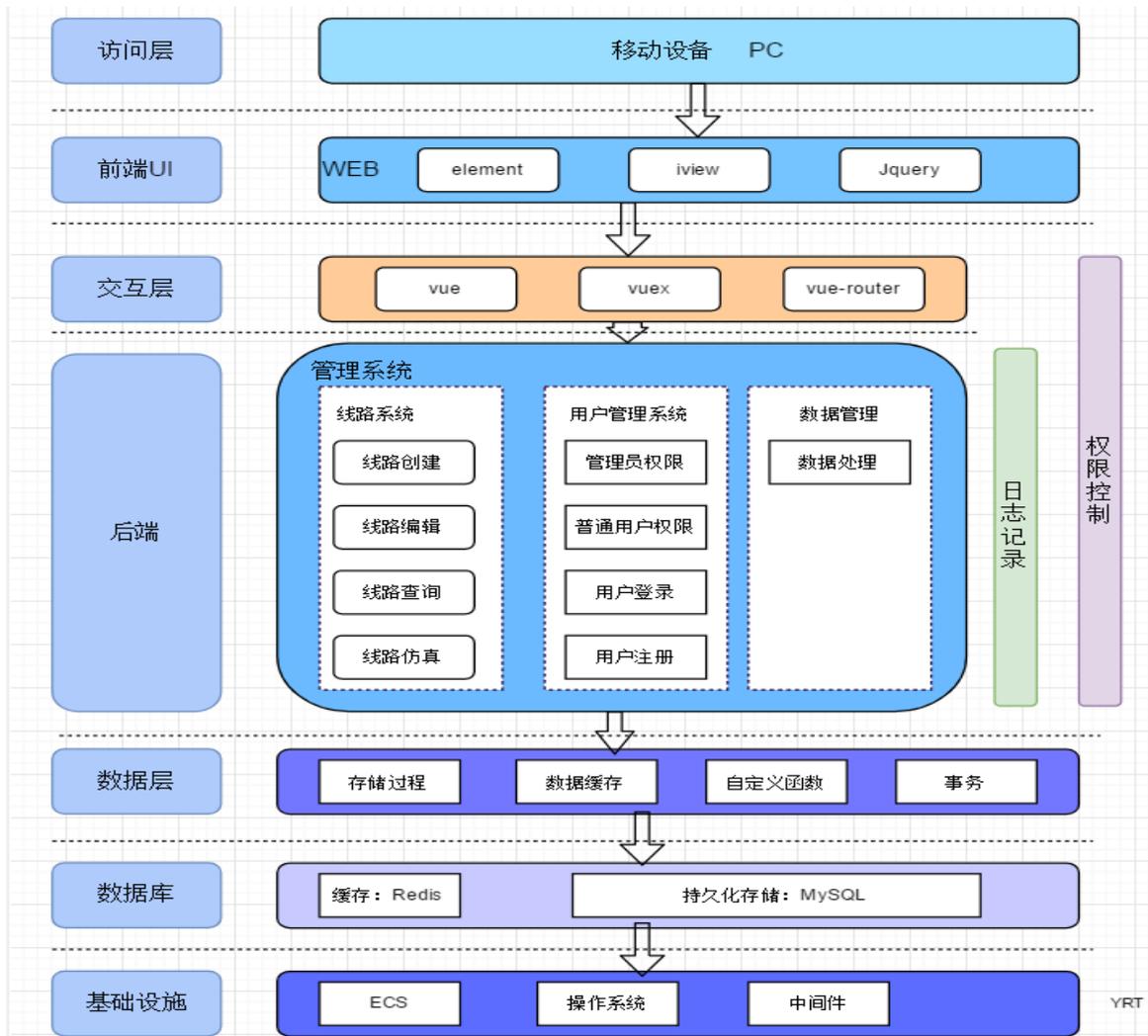


图 4.3 前后端架构总图

3.2. 系统功能模块设计

在系统主体框架中进行详细的功能模块划分。细分为了界面交互、线路系统、数据管理 4 个部分，框架的每个部分主要对应的功能模块如下表 3.1。

框架	主要对应模块
界面交互	权限控制（不同界面），登录界面、注册界面
线路系统	线路创建、线路查询、线路编辑、线路仿真
用户系统	用户管理，管理员权限
数据管理	数据处理

表 3.1 功能模块设计表

3.2.1. 界面模块概要设计

本系统主要交互在于登录界面，注册界面，菜单界面。

输入网址进入系统，首先出现登录界面，用户需要输入正确的用户名和密码，方可登录，登录界面如下图 3.2.1.1。



图 3.2.1.1 登录界面

若是用户没有账号，可以点击登录页面内的注册按钮，进入注册界面进行用户注册

功能，注册界面如下图 3.2.1.2，注册一个账号



The registration form features a large blue '注册' (Register) title at the top. Below it are four input fields: '* 用户名' (Username), '* 密码' (Password), '* 姓名' (Name), and '手机号码' (Phone Number). A blue '提交' (Submit) button is located at the bottom of the form.

图 3.2.1.2 注册界面

在登录后，加载完主界面之后，进入系统操作界面，会看到用户可操作的菜单，而在点击不同的菜单选项后，会进入不同的网页中，由于权限控制，管理员与用户界面单有所区分，管理员界面菜单功能设计如下图 3.2.1.3，点击线路查询后进入线路查询页面，输入车号进行查询，点击线路汇总后进入线路汇总页面，可看到所有线路信息，管理员可对所有线路进行编辑，点击用户管理后进入用户管理界面，可对所有用户进行管理，包括管理员，点击用户名可进行个人信息修改且可退出系统。用户界面菜单功能设计如下图 3.2.1.4 点击线路查询后进入线路查询页面，输入车号进行查询，点击线路汇总后进入线路汇总页面，可看到所有线路信息，管理员可对所有线路进行编辑，点击用户名可进行个人信息修改且可退出系统。



图 3.2.1.3 管理员界面菜单



图 3.2.1.4 用户界面菜单

3.2.2. 线路系统模块概要设计

线路创建系统主要是通过地图上画出路线，键盘输入线路信息如车号、线路编号、运行时间、距离、收益、站点，如下图 3.2.2.1 线路创建图。

* 车号	82
* 线路编号	上行途径
* 运行时间	百鸽笼公交接驳站首班车为06:20，末班车为23:00
* 距离	10.2公里
* 收益	最高票价2元
* 站点	百鸽笼公交接驳站、罗岗居委会、菁华园、华港新村、绿景山庄、草埔村口、独树村②、洪湖公园②、田贝①、东门中②、东门②、金城大厦②、金光华广场、罗湖小学、人民南



图 3.2.2.1 线路创建图

线路编辑系统主要是通过编辑已有线路，编辑车号、线路编号、运行时间、距离、收益、站点等信息，可通过线路查询，进行路线车号查询后，输入相关信息后，有更多选项，可在更多选项中选着编辑按钮如图 3.2.2.2 线路查询编辑图，进行线路编辑如图 3.2.2.3 线路编辑图。或者通过线路汇总页面直接选取线路进行线路编辑，如图 3.2.2.4 线路汇总中线路编辑图。

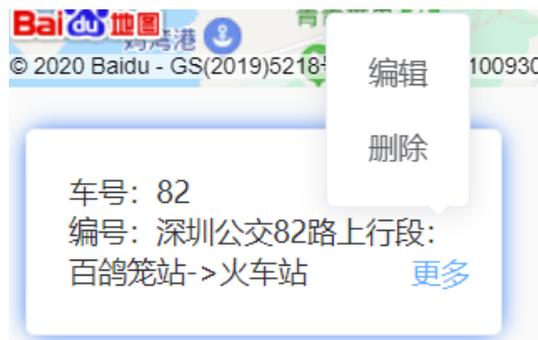


图 3.2.2.2 线路查询编辑图

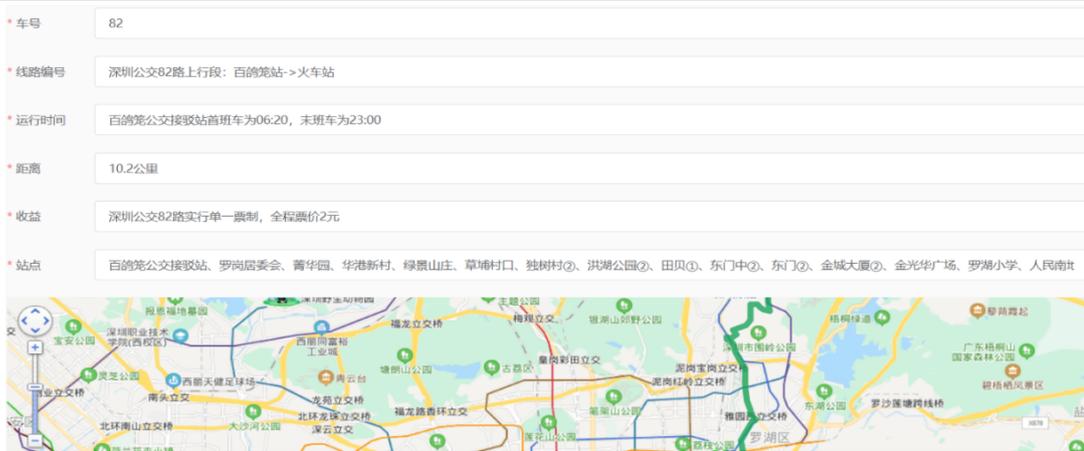


图 3.2.2.3 线路编辑图

城市公交线路仿真系统 - 颜锐涛

线路查询 线路汇总 用户管理 admin

请输入关键字

车号	线路编号	运行时间	距离	收益	备注	操作
1	深圳公交1路上行段: 布心总站->火车站	上行路线首班车为05:40, 末班车为22:30	9.02公里	深圳公交1路实行单一票制, 全程票价2元	布心总站, 布心小学, 彩世界家园, 水库, 泰宁小学, 百仕达东郡, 百仕达花园, 五方广场, 田贝四路, 田贝新村, 洪湖沃尔玛, 深圳中学, 晒布路, 东门, 金城大厦, 金光华广场, 罗湖小学, 人民南地铁站, 火车站	编辑 删除
2	2	2	2	2	2	编辑 删除
3	3	3	3	3	3	编辑 删除
82	深圳公交82路上行段: 百鸽笼站->火车站	百鸽笼公交接驳站首班车为06:20, 末班车为23:00	10.2公里	深圳公交82路实行单一票制, 全程票价2元	百鸽笼公交接驳站, 罗岗居委会, 菁华园, 华港新村, 绿景山庄, 草埔村口, 独树村, 洪湖公园, 田贝, 东门中, 东门, 金城大厦, 金光华广场, 罗湖小学, 人民南地铁站, 火车站	编辑 删除

图 3.2.2.4 线路汇总中线路编辑图

线路查询系统是通过车号进行线路查询，车号作为查询的关键字，进行精确匹配，点击查询按钮，即可查询出线路的车号，线路编号及线路信息，点击更多可对线路进行编辑及删除操作，在线路查询页面可直接添加线路，会跳转至线路添加页面。

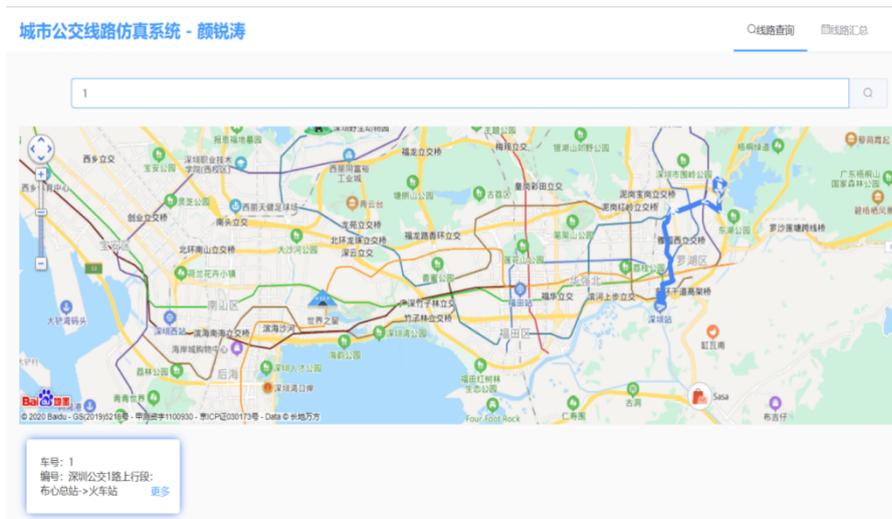


图 3.2.2.5 线路查询图

4. 功能详细设计与实现

4.1. 线路查询功能设计与实现

- 线路查询功能实现

线路查询功能实现主要体现在线路查询页面内的搜索框里，通过输入车号关键字，当按下搜索按钮，系统搜索车号后获取到线路车号，并显示出车号、线路编号、线路信息。搜索按钮图标是使用 element 的 icon 属性中的 icon="el-icon-search。本功能还可进行优化，现在只具备精确匹配功能，还未有模糊匹配功能，后续有研发需求可添加进入功能内，优化线路查询功能。

- 线路查询设计流程

线路查询功能实现流程：当输入车号关键字后，用户按下搜索按钮，后端进行查询功能，通过 getById 查询相对应的车号 ID，浏览器通过服务器发出请求，MySQL 数据库服务器返回相对于数据，前端再通过 handelSearch 函数如下图 4.1.1，搜索相对应的车号关键字，从而获取到车号，系统就会从 JSON 里获取到保存好车号相关信息，在百度地图上显示出线路仿真，且在页面内显示出车号、线路编号。

```
handelSearch() {
  this.search = this.searchTemp
  let param = {
    name: this.search
  }
  if (!this.search) {
    interaction
    return
  }
  route.routePage(this, param, pageNum: 1, pageSize: 100).then(res => {
    if (res) {
      let data = res.data
      this.showAllOverlay()
      this.form.point = JSON.parse(data.point);
    }
  })
},
```

图

4.1.1 handelSearch 函数

同时通过点击更多按钮，触发 more 函数，调用 element 的 el-dropdown 设置 el-dropdown-item 属性，出现下拉框，进行操作选择，能对线路进行编辑或删除功能，点击编辑按钮就会跳转至线路创建页面，进行数据编辑。点击删除功能，系统会弹出确认删除框，咨询用户是否进行删除确定或删除取消。

4.2. 线路创建设计

● 线路创建基本设计

公交线路信息有车号，线路编号，运行距离，运行时间，收益，经过的站点，公交线路在地图上的路线，这是一条公交线路的基本信息，用户输入完相应的信息后，点击提交，即可完成一条线路的创建，可在线路查询及线路汇总查看线路信息。

● 线路创建设计流程

用户线路创建的时候，需要键盘输入车号，线路编号，运行距离，运行时间，收益，经过的站点。当用户点击网页内的添加按钮，`@click="linkTo"`，`linkTo` 函数会使系统自动使得网页跳转至线路创建页面，前端通过使用 `element` 的 `el-input` 创建数据输入框，用户输入玩基本信息后。

接着在百度地图上创建公交路线，利用百度地图 API 调用百度地图，`handler` 函数实例化百度地图，路线的实现是通过调用百度地图 API 的属性 `Polyline` 折线以及 `Overlay` 覆盖物，实现添加覆盖物至地图上，同时记录下 `point` 经纬度，形成公交路线。公交路线在地图上通过右键点击地图 2 点，形成一条 `Polyline` 折线，左键控制移动地图显示范围，创建折线后，可对折线进行调整，将鼠标移动至折线上 `polyline.addEventListener('mouseover', (type, target, point, pixel))`，鼠标进入曲线可以进行编辑，左键点击拖动折线上的点，调整公交路线，鼠标移出折线后，返回 `polyline`，并且保存相对应的 `point` 经纬度，如下图 4.2.1 路线创建图。

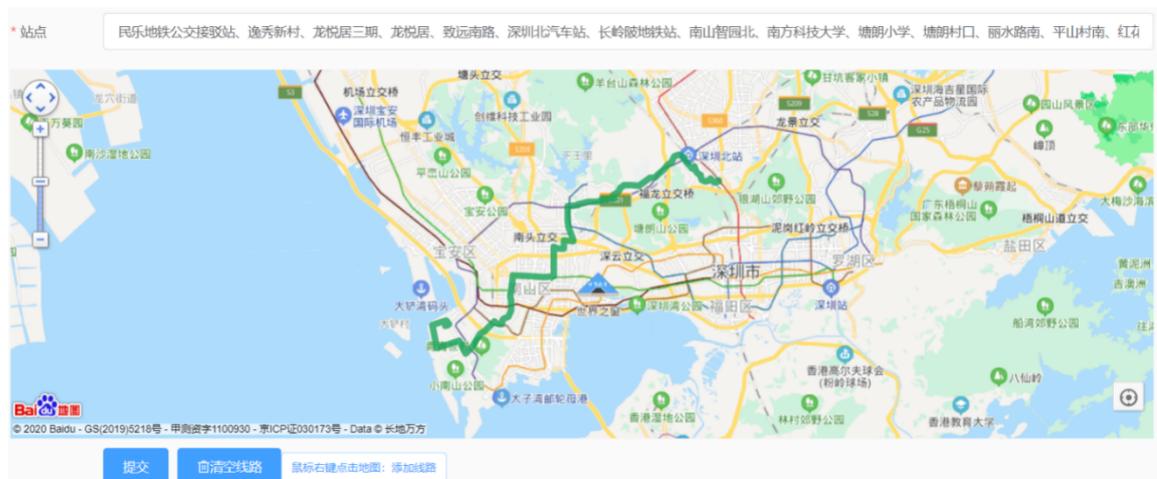


图 4.2.1 路线创建图

清空线路按钮是通过 `click="clearOverlays"` 函数，点击按钮后，进行线路清除即清除覆盖物，清除所有覆盖物，所以设定了一个提示框函数 `interaction.showModel.confirm('确认清空所画路线?', '提示')`，询问用户是否清空线路，避免用户误删线路。效果如下图 4.2.2 清空线路提示图。

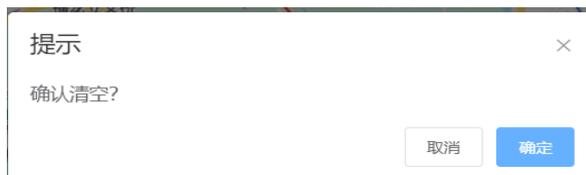


图 4.2.2 清空线路提示图

线路信息填写完毕，地图上的路线也画完后，即可点击提交，提交线路，点击提交

按钮后，会有提示框提示用户线路创建成功，后端就会通过 public Object save(Route route)函数，保存用户创建的路线，同时保存数据至数据库内，永久存储，根据需求前端也会有一份缓存数据，用户需填写所有线路信息及画出路线，否则系统会提示缺少了什么，添加后才能点击提交按钮，创建线路，至此线路创建功能就已经完成了。

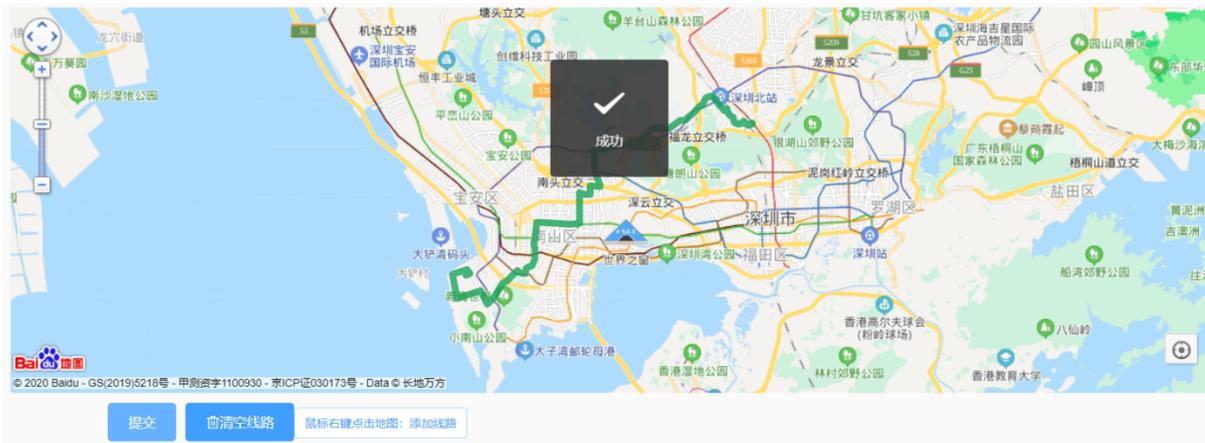


图 4.2.2 路线创建成功图

- 功能可优化点设计

本功能可优化的点就是地图上画路线的功能，右键添加路线的同时可以获取到地图上的信息，还有就是路线清除功能，如果设置成 2 个按钮，清空所有路线和清除上一段路线，路线清除功能也会得到更好的完善，后续笔者技术有所提升还会继续研究如何优化该功能。

4.3. 线路编辑设计

- 线路编辑基本设计

线路编辑功能主要是通过点击编辑按钮@click="edit(index)"相当于进入线路创建页面，系统获取原有数据后，进行原有线路上的数据编辑。

- 线路编辑设计流程

进入页面方式：

1. 在线路查询页面内，点击线路信息内的更多选项@click="edit(index)"，弹出下拉框，出现编辑选项，点击进入线路编辑页面。

2. 在线路汇总页面内，查看到线路信息，在操作一栏可见编辑选项，点击跳转至线路编辑页面（受权限控制，普通用户无法编辑管理员创建的线路）

进入页面后，系统通过 ID 确认，获取线路信息，

```
if(id){  
    this.id = id  
    route.routeGetById(this, id). then}
```

确认线路 ID 后，通过 data 显示线路相关信息，如地图上的路线经纬度：
this.form.point = JSON.parse(data.point)，车号，线路编号等信息。

用户直接键盘输入修改线路信息即可，地图上的路线可利用清空线路再次规划路线或移动鼠标至路线上，进行拖拽修改。以上操作与线路创建时相同，本功能实现点在于点击编辑后，获取原有数据，与后端数据进行交互。信息都填写完毕后，方可点击提交按钮，后端函数 public Object updateById(Route route)运行后，线路信息保存至数据库内，，用户可以直接查看修改后的线路信息。如下图 4.3.1 原有线路数据图、图 4.3.2 修改后线路数据图。

车号	线路编号	运行时间	距离	收益	备注	操作
0	0-线路编辑	12	212	54654	4524	编辑 删除

图 4.3.1 原有线路数据图

车号	线路编号	运行时间	距离	收益	备注	操作
0	0-线路编辑修改后	1431413	413543	3425	125431	编辑 删除

图 4.3.2 修改后线路数据图

4.4. 线路汇总设计

- 线路汇总基本设计

本功能设计初衷是为了方便用户能更好更直观的看到公交数据并进行查询，编辑等

操作，本功能是通过调用 Element-UI 的组件 el-table 来作为表格的显示，从而展现多条数据以及可对数据进行对比等自定义操作。还有调用 Element-UI 的组件 Pagination 分页，是为了优化数据展示效果以及分化数据，更便于展示数据以及进行对比。

- 线路汇总设计流程

首先创建基础的 Table 表格<el-table :data="pageInfo.records" border style="width: 100%">，其中的 data 通过 pageInfo.records 获取，获取函数如图 4.4.1 表格数据添加图。

```
return request.get(url, data).then(res => {
  let resData = res
  if ('success' == resData.status) {
    //如果大于1执行添加
    resData.data.records.forEach(i => {
      try {
        i.point = JSON.parse(i.point)
      } catch (e) {
        console.error(e)
      }
    })
  }
  let pageNum = resData.data.current;
  if (config.pageConcat && pageNum > 1) {
    /** 添加 ***/
    let list = _this.pageInfo.records.concat(resData.data.records);
    // console.log(list)
    resData.data.records = list;
    /** 添加 ***/
  }
})
```

图 4.4.1 表格数据添加图

获取到数据后，表格基本能显示出来了，现在需要设计分页功能，同样也是调用 element 的组件 el-pagination 设置，随着数据的增加，数据条数的变化，以及分页可选择页数。利用其中的 size-change 以及 current-change 事件修改页码大小和当前页变动时候触发的事件。current-page 表示当前页数，支持 .sync 修饰。显示效果如下图 4.4.2 分页功能显示图。

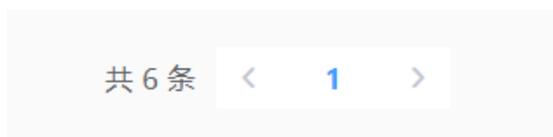


图 4.4.2 分页功能显示图

4.5. 线路仿真设计

- 线路仿真基本设计

本功能为本系统的难点之一，实现功能效果：线路查询中输入车号，搜索出来线路，地图上的路线会有辆小车循着轨迹运动。

- 线路仿真设计难点及尝试方案

设计本系统的时候，笔者想先实现画路线再实现小车循着轨迹运动，可当设计好画路线的功能后，尝试加入小车循迹运动，出现了BUG，经过多次调整以及与导师沟通后，尝试多另外几种方法。

方法一：使用Ecahrts插件来实现车辆轨迹运动，通过调用echart以及百度地图API，实例化echart设置属性后，笔者发现原本设置好的路线显示不出来了，数据读取不到，多次调整后，线路能显示出来，可无轨迹可视化，仍然是原本的静止效果，这个方案先暂定，尝试别的方案先。

方法二：百度路书轨迹播放，使用一个播放按钮来实现轨迹运动，配置文件设置完成后，路线也显示出来了，可是调试运行的时候，一直报报错BMapLib is not defined，查询了相关资料后，重新调整配置文件，仍然不可行。

由于时间上的问题以及笔者对于vue刚入门不久，本功能先跳过，简化实现，等有更好的方法或者学习到如何实现后再来完成该功能。

- 线路仿真暂定解决方案

无法实现小车循迹运动，先实现路线有方向及起终点，选择在线路查询后，在地图路线上线路箭头，以表示路线的运动。可对箭头进行大小和粗细调整，如下图4.5.1路线实现图

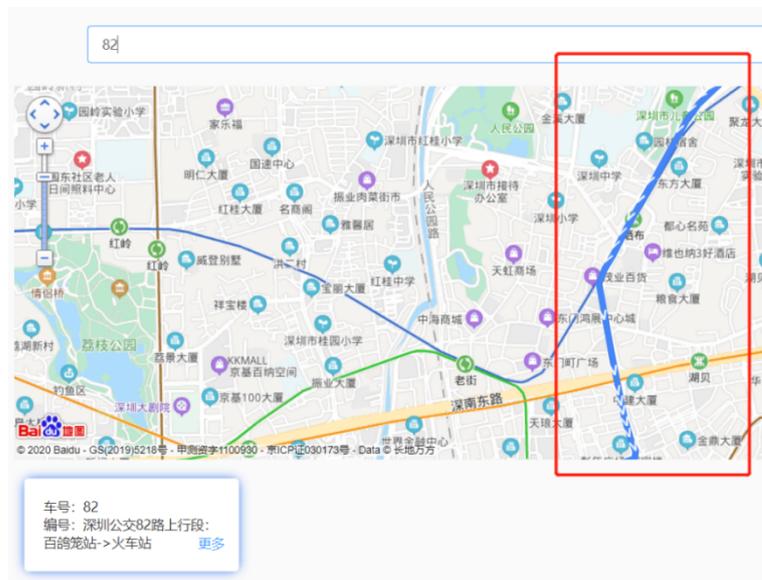


图 4.5.1 路线实现图

4.6. 登录与注册功能设计

- 登录与注册基本设计

本系统另一个难点就在于登录与注册功能，登录功能实现效果为：当用户输入账号及密码，若输入正确账号密码则成功登录系统，系统弹出登录成功提示窗

，反之则会弹出提示窗输入错误，提示用户账号或密码错误。若是用户未注册账号会提示账号或密码错误，所以需要进行用户注册功能。

注册功能是未做限制注册的，任何用户都可以进行账号注册，注册成功后即可登录系统，进行系统操作，不过只能对本用户所创建的线路进行操作。

- 登录功能设计流程

1. 登录成功设计流程

首先需要做好准备工作，配置 vuex，使得页面刷新时将 vuex 里的信息保存到 localStorage，这样子刷新页面就无需再次登录如下图 4.6.1 刷新配置，

```
storage.getSession("state") && store.replaceState(storage.getSession("state"));
//在页面刷新时将vuex里的信息保存到localStorage里
window.addEventListener( type: "beforeunload", listener: ()=>{
  storage.setSession("state",store.state)
})
```

图 4.6.1 刷新页面配置

准备好即可写 Login 的页面及函数，还需要设置监控，同时还需要修改主页内的菜单栏，用户登录后，显示出用户名称，最后就是与后端进行交互，进行用户名及密码验证，基础函数设计完毕后。

由于需要进行权限区分，还需要设置多一个 .java，去分辨用户为普通用户还是管理员，对 UsernamePasswordToken 进行了简单的封装，用于判断该用户是需要用哪个 realm 进行登录，还有获取当前用户授权信息，进行授权处理，区别开普通用户只能操作普通用户创建的线路。如下图 4.6.2 身份认证函数。

```
@Override
protected AuthenticationInfo doGetAuthenticationInfo(AuthenticationToken token) throws AuthenticationException {
    // 获取用户的输入的账号.
    String username = (String) token.getPrincipal();// 获取身份
    logger.info("Authentication success : " + username);
    UsernamePasswordToken usernamePasswordToken = (UsernamePasswordToken) token;
    // 密码就是凭证
    char[] passwordChar = usernamePasswordToken.getPassword();
    String password = new String(passwordChar);// 获取到密码
    // 设置为肯定登录成功
    SimpleAuthenticationInfo authenticationInfo = new SimpleAuthenticationInfo(username, // 用户名
        password, // 密码
        // ByteSource.Util.bytes(user.getId()), // salt=username+salt
        getName() // realm name
    );
    return authenticationInfo;
}
```

图 4.6.2 身份认证函数

2. 登录失败设计流程

本功能的登录失败目前做了两种失败提示区分，第一种：当用户输入错误的账号和密码，系统则提醒用户账号或密码错误；第二种：当用户输入正确的账号后，输入错误的密码，系统则提示用户密码错误。

用户输入账号和密码错误，点击登录按钮后，数据会传到后端进行验证及判断

，如果账号和密码都不正确的话，弹出提示框 用户账号或密码错误，请重新输入，调用到 public static BaseResult fail(String msg)函数，返回一个 msg：“账号或密码错误”。

用户输入正确的账号后，密码输入不正确，点击登录按钮后，后端会验证到用户名存在且正确，在验证密码，如果密码不正确，网页会弹出提示框，提示用户密码错误，请重新输入。登录 Login 类设计如下图 4.6.3 Login 类设计

```
@PostMapping("/login")
public Object login(User user) {
    String username = user.getUsername();
    String password = user.getPassword();
    if (StringUtils.isBlank(username, password)) {
        return BaseResult.fail("账号或密码为空");
    }
    user.setPassword(null);
    User selectOne = userService.lambdaQuery().eq(User::getUsername, username).one();
    if (selectOne != null) {
        if (!password.equals(selectOne.getPassword())) {
            return BaseResult.fail("密码错误");
        }
        Subject subject = SecurityUtils.getSubject();
        Session session = subject.getSession();
        UsernamePasswordByUserTypeToken usernamePasswordToken = new UsernamePasswordByUserTypeToken(username,
            password, ShiroEnum.SIMPLE_USER.getCode());
        // 只要进了这里，肯定登录成功
        subject.login(usernamePasswordToken);
        session.setAttribute(ShiroEnum.SESSION_USER_INFO.getCode(), selectOne);
        Map<String, Object> returnMap = new HashMap<>();
        subject.hasRole("admin");// 将权限加入缓存
        AuthorizationInfo authorizationInfo = simpleUserRealm.getAuthorizationCache().get(subject.getPrincipals());
        if (authorizationInfo != null) {
            returnMap.put("roles", authorizationInfo.getRoles());
        }
    }
}
```

图 4.6.3 Login 类设计

- 注册功能设计流程

用户未注册账号，在进入系统界面时，点击注册按钮，系统跳转至注册页面，在页面输入用户名、密码、姓名、手机号进行注册，手机号需输入正确的手机号，否则系统会提示用户请输入正确的手机号，系统会验证手机号码是否正确，调用到 isPhone 函数如下图 4.6.4 isPhone 函数，用户名及密码都有数字长度限制，需长度在 3 个字符以上。

```
export const isPhone = phone => {
    var reg = /^[1][3,4,5,7,8][0-9]{9}$/;
    return reg.test(phone)
}
```

图 4.6.4 isPhone 函数

后端会进行账号是否存在检测，用户填写完成后，点击注册按钮，即可完成注册，就可以登录系统进行使用了，后端就需要完成数据提交及用户创建的功能，register 类设计如下图 4.6.5 register 类设计图，后台也要设置好一个注册接口 filterChainDefinitionMap.put("/user/register", "anon")，到这里注册功能算是完成了，接下来就是简单的数据调试下，是否有保存进数据库内，调试成功即可完成注册。

```

@Transactional
public Object register(User user, String roleName, String roleSn) {
    String username = user.getUsername();
    String password = user.getPassword();
    if (StringUtils.isBlank(username, password)) {
        return BaseResult.fail("账号或密码为空");
    }
    User userOne = userService.lambdaQuery().eq(User::getUsername, username).one();
    if (userOne != null) {
        return BaseResult.fail("该用户已存在");
    }
    user.setId(UUIDUtils.getUUID());
    userService.save(user);
    // 添加角色
    setRole(user.getId(), roleName, roleSn);
    return BaseResult.success(user);
}

```

图 4.6.5 register 类设计图

- 登录与注册功能注意事项

跨域问题解决方案

跨域是什么呢？跨域就是一个网域向另一个网域进行资源交互，因此跨域访问违背了同源策略（两个网域的“协议+域名+端口”相同），才会当请求不在同一域名下的资源文件时出现了 No 'Access-Control-Allow-Origin' 错误。在本次设计过程中的同样也需要解决跨域问题，要访问 8088 端口的服务器，会报错，是因为 VUE 需要占用 8080 端口。这个问题在前端或后端处理都可以。在本次设计中我是在后端处理的，使用 cors 解决跨域使用 json 返回数据给前端，设置了 CorsConfig 类如下图 4.6.6 CorsConfig 类。

```

public class CorsConfig {

    @Bean
    public CorsFilter corsFilter() {
        UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();
        source.registerCorsConfiguration( path: "**", buildConfig());
        return new CorsFilter(source);
    }

    private CorsConfiguration buildConfig() {
        CorsConfiguration corsConfiguration = new CorsConfiguration();
        // 1允许任何域名使用
        corsConfiguration.addAllowedOrigin("**");
        // 2允许任何头
        corsConfiguration.addAllowedHeader("**");
        // 3允许任何方法 ( post、get等 )
        corsConfiguration.addAllowedMethod("**");
        // 允许保存凭证
        corsConfiguration.setAllowCredentials(true);
        return corsConfiguration;
    }
}

```

图 4.6.6 CorsConfig 类

4.7. 权限控制设计

- 权限控制基本设计

由于系统需要区分开普通用户及管理者的使用情况，故设计了权限控制功能，用户注册后为普通用户，如需升级为管理者，需管理者给予权限后，才可升级为管理者，只

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要
下载或阅读全文，请访问：

<https://d.book118.com/505110002104011131>