公交查询系统

# 摘要

在二十一世纪，公交车是人们赖以维持出行的交通工具，公交车的出现以及广泛应用，让人们的生活也变得更加便捷。随着社会的发展和进步，越来越多的人需要出行，公交也在人们的日常生活中占取着越来越重要的作用。与此同时，有出行需要的人们都迫切地希望能够掌握准确详细的公交信息，面对纷芜繁杂五花八门的公交线路，人们往往要花费一些时间查找公交车的信息，出行乘坐公交车前必须要了解公交车的是否能到达目的地，以及哪一路公交车能满足需求。

本文针对这种情况，撰写了公交查询系统。本设计旨在提供给广大出行者线路查询、站点查询以及换乘查询功能，并且还有留言功能用以采纳建议以完善系统，方便人们查询公交信息，规划好出行路线，使出行更加方便快捷。公家查询系统的实现采用了python编程语言以及MYSQL数据库技术。本论文主要介绍公交查询系统中已实现的功能以及关键技术。

**关键词**：公交查询、交通、python、数据库

**Public Transport inquiry system**

# Abstract

In the twenty-one th century, the bus was the only means of transportation that kept people going,The appearance and wide application of the bus make people's life more convenient. With the development and progress of society, more and more people need to travel, public transport also plays an increasingly important role in people's daily life. At the same time, people who need to travel are eager to have accurate and detailed bus information. Faced with the multifarious bus routes, people often spend some time to find bus information, before traveling by bus must know whether the bus can reach the destination, and which bus can meet the demand.

In view of this situation, this article has written the public transportation inquiry system. The purpose of this design is to provide the general traveler route inquiry, the station inquiry as well as the transfer inquiry function, and also has the message function to adopt the suggestion to perfect the system, facilitates the people to inquire the public transportation information, the plan good trip route, to make travel easier and faster. The implementation of public query system is based on python programming language and MYSQL database technology. This paper mainly introduces the functions and key technologies of public transport query system.

**Keywords:** Public Transport Query, traffic, python, database

# 前言

现如今人们的生活正发生着翻天覆地的变化,在让世界瞩目的中国基建速度中,尤其不断增多的四通八达的公路最为出彩。城市的发展也在推动着交通的发展，公交运行线路越来越多,人们出行乘坐公交的需要也在急遽增加。交通是城市的命脉,而公共交通具有覆盖面广、容量大和方便快捷等优势[1],在城市的经济飞速发展下,城市的规模变得更庞大了,相应的城市公交系统也变得更复杂了。大规模的新建住宅区与四通八达的公交线路日益增多,一方面让人们的生活质量有了质的飞跃,另一方面也让人们的出行越来越便捷和便利。经济的飞速发展导致我国机动化和城镇化进程不断加快,据统计,截止到2012年底,我国机动车保有量已达到2.4亿,机动车驾驶人数已达到2.6亿[2].但是由于城市规模的不断扩大,人们往往很难选择出行路线。即使对一个生活在本地城市很多年的人来说,对于五花八门的乘车线路有时也会思索良久。更何况是对于在城市里初来乍到的人们来说,短时间内对于熟悉公交路线是很难的。所以就需要一个来满足人们短时间内找到最佳线路的公交查询系统,假如有一个这样的系统,一个既能可以用来查询公交车的站点和线路,又能够用来换乘的系统,这个系统能够让乘客不再盲目的等待，而是让乘客可以根据公交车次的情况而出门，从而大大的节省了乘客的时间。有了公交查询系统,不仅极大程度上方便了人们的出行,同时更是侧面表现出了一个地区的智能公共交通系统的高水平与这个地区的数字化和信息化水平。

如今,我国许多大城市都有交通阻塞的情况在,这不仅仅延误了人们的上下帮时间。也影响着城市的蓬勃发展。因此为了有效改善公交车的拥挤状况,适当增加出发次数,以节约乘客时间,所以就需要为人们提供最便捷的公交线路,这将很大程度节约了乘客的时间。而公交查询系统,使人们可以实时查看公交情况，从而为乘客规划处最快的出行路线。

# 第1章 绪论

## 1.1 课题的目的与意义

随着科技日新月异的发展，人们越来越喜欢出行，不管是出门走亲访友还是去各地旅游，都离不开发达便捷的交通工具。当人们来到陌生的城市时，很重要的一点就是要规划好出门的乘车路线。于是就有许多方便人们出行的应用也应运而生，城市公交运输的特点不仅有覆盖面广泛，同时也有便于经济发展的特点。就目前而言,绝大多数出行者出行的首选方式还是公交车。越来越多的国家和地区都致力于发展当地的公共交通事业努力提升公共交通的设施与服务。同时根据人口密度的分布合理调度交通路线的安排。面对发展得如火如荼的公共交通事业，如果可以提供一种服务,为有出行需要的市民或者是到当地旅游、出差、念书等急需要了解本地道路的人们当地公交路线的信息，这就能节省人们查找询问公交路线的时间，省去诸多麻烦，让人们的生活和出行更加便利快捷。与此同时信息明确的公交线路方案能够让人们出行时优先选择乘坐公交，这就减少了道路上不必要的交通流量，可以缓解市区的交通拥堵，大大提高公众的出行效率，并可以促进市区的发展。在世界各地，公交车是城市的活力，通过四横八达交纵相错的路线将人们送往不同的目的地。

对于要出行的人们来说，需要这样一款应用来查询自己出门将要乘坐的公交车的停靠点、所经路线及到站时间。现如今城市的发展正如火如荼，公交路线不断增多，并且公交路线时有增改，有许多市民会对各条公交线路的信息不清楚或者掌握得不够准确，有了公交查询系统，人们就能更好的知悉公交车的信息，外地的人不用担心自己人生地不熟而不知道坐哪一趟车，外出久归的人不用因公交线路发生改变而苦恼，有乘坐公交车需要的人也不必再担心是否会因掌握不好时间而错过公交车。故此，公交查询系统的产生极大程度上方便了市民的出行。有了正确的乘车路线，就能为用户节省许多时间，提高办事效率，方便人们出行。

在现在大力倡导绿色出行的时代背景下，出行选择公共交通工具是大势所趋，各个地区的公共交通体系都正在被完善和改进。近年来，互通技术的发展一直如火如荼，在这个发展趋势下，互联网已经在人们的生活中无处不在了。人们越来越习惯于网上冲浪,从网络上获取相关资讯，于是公交查询系统在这个时代背景下应运而生,它具有站点查询、换乘查询线路查询以及线路查询这些很实用的功能。只要进入了公交查询系统，就能够很快获悉公交线路的具体情况，也能够立马查询到公交站点的详细信息，除此之外还能查询到换乘的公交线路。网络技术的发展给公交查询系统注入了一股新的活力和力量，根据信息化技术建立公交查询系统。使用互联网手段不仅能加强公交公司的管理，还有利于公交行业服务水平的提高，在激烈的现代化企业竞争中能够脱颖而出，始终占据有利位置。

许多城市的公交系统十分庞大，错综复杂，市民们在自己的城市出行，有时也需要通过查询才能出门，更何况是来自外地的人们了。因此，有一个掌握各路公交信息的公交查询系统就显得十分重要了。

## 1.2 公交查询统的现状和前景

相对于发展国家来说，我们国家的公交查询系统尚且现阶段还在发展中。世界上有很多国家都十分重视城市公交运输行业的发展，他们大力发展公交交通运输，投入了大量的人力和物力。在保持大力发展的力度下，这些国家基本上建成了完善的互联网公共交通运输系统。这样很大程度上改善了交通环境，极大地方便市民的出行。国外有很多国家和地区发展公交运输起步早，发展速度也很快，很早就重视公共交通的规划。通过大力投资建设互联网公交系统与现代化通信技术，已经形成了适应本地情况的成熟的公交管理体系。经过多年的发展，积累了相当多的日常交通数据，在研究了公交调度等关键基础理论后，他们在公共交通系统中融入计算机网络与信息化技术，在经年以往的发展下，积累了很多交通发展经验，基本完成了公交管理的信息化。

近些年来，我国开始着力于对公交查询系统的研究,对我国公共交通事业加大发展力度。新世纪以来，由于城镇化与城市化的发展，离开原先生活的农村地区进而选择定居城市的人们越来越多，这股人流的浪潮也带来了相关的问题，许多之前没见过的情况也随之而来，比如：城市的道路一天天变宽变大，然而交通却是一天天的变得拥堵；生活质量一步步的变好，而居住的自然环境却是一步步的恶化。随着人们的经济状态变好，对生活的追求也在提高，于是有能力购置私家车的人们也原来越多。可是人们的出行却并没有因此变得更加畅通，因为汽车等私人交通工具的增多往往也会导致交通运输的压力，于是便于人们出行的重担还是落在公共交通上。

尽管我国城市化速度快，占地面积大，但规划不明确和人口密度大的问题仍然存在。于是就要通过积极发展公共交通来解决这些问题，无论时代怎么变迁，城市公交始终是最高效的交通运输方式。因为各种原因，我国的公共交通服务水平还处在一个较低阶段，公交的运行效率不高,发展总体来说相对滞后。我国公交查询系统的建立尚且还处在一个起步阶段,但是市民们对公交出行的需求量却越来越大。这种不对等的供求关系迫切的要求我们加快公交运输系统的发展，扩大人们查询公交信息的方式、来源和手段。由于目前我国的城市公交信息系统发展还不够完善，广大乘客能够获取公交信息的途径还不够多。这对大出行人士来说十分不便。基于这样的环境背景下，以加快发展和完善城市公交信息化为目的，搭建一个公交查询系统是很有必要的。

## 1.3 课题的主要内容及要求

根据对调查公交车具体情况的调差研究，摸清当地公交线路、站点安排以及运行时间间隔和频率，对市民日常生活中选择的乘公交路线，换乘的习惯等情况做好攻略，规划好公交查询系统应该具有的功能和要实现的目标。对公交查询系统进行研究和设计，了解整个软件的设计和开发的过程。本系统包括了六个模块。这六个模块分别是：系统登录模块、管理员模块、公交站点查询模块、公交线路查询模块、公交换乘查询模块以及留言模块。

1. 系统登录模块：公交查询系统的登录有管理员登录和普通用户登录，他们分别有自己的用户和密码，用户选择身份后，对于登录成功的普通用户，登录后进入用户个人界面进行相应操作，对于登录成功的管理员，登录系统后进入管理员界面进行相应操作。如果输入的密码和账号是错误的，系统会给予出错提示，让用户重新进行登录。
2. 管理员模块：管理员登录进入管理员界面后，对路线和站点进行增加和删除，对路线进行管理，将站点归并到途径这些站点的公交路线中去，另外管理员还可以在此界面查看普通用户登录系统的账号和密码。查看普通用户的留言信息。

（三）公交站点查询模块：用户根据自己的需要输入站点名称，可以查询到经过这个站点的公交线路信息，得到乘车方案。

（四）公交线路查询模块：用户输入要查询的线路名称，可以得到包含在该线路内的按顺序经过的所有站点信息。

（五）公交换乘查询模块：用户在公交查询系统中输入起始站点和终点站，可以查询到在这两个站点间的公交换乘方案。

（六）留言模块：用户可以在此模块留下自己的意见与建议，以供公交管理人员根据用户的建议将系统更好的完善。

# 第2章 开发工具及相关技术

## 2.1 常用开发工具

### 2.1.1 python平台简介

Python是高级程序设计语言，它带有动态语义。同时它又是解释型的，面向对象的，自从Guido van Rossum在20世纪90年代初创造这门语言以来,他的支持者就一直在稳步增加,而且近些年来,社区对他的兴趣也日益浓厚[12]。Python存在分别基于不相同语言而开发的多种解释器，每个解释器都有各自的特点，但它们还是都能正常地运行Python代码。Python不仅有开发周期短、使用快捷的优点，并且python还能够担任好脚本语言的工作，Python是一种很便于使用的shell脚本语言，同时它也是一种控制语言，只不过python更加擅长快速而灵活的开发模式，python扮演了非常多的角色[5]，在德国举办的CeBIT2009大会上，Python获得了最佳开源编程语言奖，这一奖项直接肯定了Python在提升语言功能性和易用性方面一直以来做出的贡献[14]。

Python的特点在于它拥有简洁的语法并且不需要编译，而且拥有最优化的开发速度。与其他编程语言相比，选择python的话会大大缩短开发时间，而且会大大提高开发效率。值得一提的是用python开发出的软件运行效率也非常高。相对于Java和C++来说Python 更加简单也更加易于使用，其相对于visual basic来说又跨平台性能更为出色，又比 Ruby更加成熟和更具可读性,比 PHP 更通熟易懂有更宽广的用途,比 Perl 设计的语法更加方便简洁等等，而随着Python越来越受到开发者的欢迎,现在Python已经拥有了Jython, IronPython等其他实现[6]。

Python解释器主要有以下几个:

(1）CPython：这个解释器是官方版本的：CPython。它之所以被命名为CPython。是因为它是一个用C语言开发的解释器。需要启动CPython解释器时，就在命令行下运行python即可。CPython是目前来说使用最广且很有发展前景的Python解释器。

（2）IPython:IPython是基于CPython之上的一个交互式解释器，换言之IPython仅仅提高了交互方式，仍然与以CPython相同的方式执行Python代码。CPython用>>>作为提示符，而IPython用In[序号]:作为提示符。

(3)PyPy：PyPy是另一种类型的Python解释器，对Python代码动态编译时采用的是JIT技术，它以提高执行速度为目标。

### 2.1.2 pycharm介绍

PyCharm是一种Python IDE，带有一整套可以帮助用户在使用Python语言开发时提高其效率的工具[7]，它有调试、语法高亮的功能，也有Project管理、代码跳转的功能，同时还可以执行智能提示，自动完成，单元测试，版本控制等功能。此外，此IDE还提供了其他高级功能。以便能更好地来支持Django框架下的Web开发。曾经著名的JetBrains打造了一款实用的Python IDE，它就是PyCharm。同时JetBrains还打造出了VS2010的重构插件Resharper。该插件可以支持Google App Engine，而PyCharm也支持IronPython。PyCharm的这些功能不仅是开发人员的有力工具，也非常适合刚入门的人。PyCharm不仅提供常规IDE的功能，还提供可用于Django开发的其他一些出色功能。PyCharm还支持Google App Engine和IronPython。它的功能是：

1. 编码协助：由于它带有一个带编码补全和代码片段，不仅有助于用户完成编码工作时更快更轻松，也有支持代码折叠和可配置的编辑器这样的功能。此外它还具有分割窗口的智能。

（2）项目代码导航：使用此IDE，用户可以立即在两个文件之间进行导航。它的用法也可以遍历类层次结构。学习了如何使用提供的快捷键的用户也可以更快，更方便地使用它。

（3）代码分析：用户可以进行智能检测等手段让编码得到优化。

（4）Python重构：此功能使用户可以在项目范围内更简洁地重命名。

（5）Django支持：用户可以使用HTML，CSS和JS编辑器更快地开发自己的网站。此外，Djang框架还可以支持CoffeeScript，Mako和Jinja2。

（6）Google App Engine支持：您可以使用Python 2.5或2.7操作环境，并且可以在执行常规程序扩展时执行Google App Engine应用程序开发。

（7）集成的版本控制：集成的VCS用户界面（在Mercurial，Subversion，Git，Perforce和其他SCM上可用）使您可以查看登录，记录，拆分和合并功能。

（8）图形页面调试器。

（9）集成单元测试。

（10）可定制和可扩展。

### 2.1.3 Mysql介绍

MySQL是瑞典MySQL AB公司开发的一个关系型数据库管理系统，目前属于Oracle 旗下产品[8]。它的优点是体积不大、免费使用，开放源码。由于总体拥有成本也不算高，所以在Internet上的中小型网站中MySQL得到了广泛应用。公交查询系统选择将Python和MySQL作为网站数据库。MySQL是一个很受欢迎的关系数据库系统。人们现在对数据存储的要求越来越高，必须不能浪费每一个能存储信息的机会，要储存有意义的数据集合，为了整理分析数据之间的关系，数据库是一个不可或缺的有效手段。

每个数据库都有API，在对数据进行操作时就要用到API。如果将数据直接放在文件里，那么文件中的数据就不能快速读取或写入。因此，我们目前使用关系数据库管理系统（RDBMS）来存储和管理大量数据。关系数据库不是存储所有数据的大型仓库，但是数据存储在各种表中。该关系数据库基于关系模型。 在数据库中处理数据时，它是基于数学概念或方法（例如集合代数）的。关系数据库管理系统RDBMS(Relational Database Management System)的特点如下：

（1）数据以表格格式显示。

（2）各种记录名称存储在每一行中。

（3）与记录名称相对应的数据字段存储在每一列中。

（4）表格由许多行和列组成，

（5）数据库由几种格式组成。

在本设计中，将公交车辆信息保存在数据库中，利用相应的数据库工具对存储的数据进行操作。

### 2.1.4 Django介绍

设计该系统为了减少花费，所以采用快速而免费的Web开发技术。传统的重量级开发技术ASP.NET和J2EE显然不适合轻量级的快速开发。故采用基于Python开源的Django框架[13]。Django应用开发遵循MVC模式[15]，Django是一个Web应用框架，它是开放源代码的，由Python语言写成。Django采用的是MVT（模型M，视图V和模版T）的框架模式。开发Django的最初目的是管理网站。Django是从真实世界的应用中成长起来的，它是由堪萨斯州 Lawrence 城市中的一个网络开发小组编写的[9]，在BSD许可证下，它发布于2005年7月。一个吉普赛爵士吉他手Django Reinhardt命名了这套框架。Django同时也是一个基于MVC构造的框架。MVC作为一个框架模式，强制性的使应用程序的输入、处理和输出分开[10]。还未引入MVC模式模型得实惠，WEB服务器大多为两层架构，其中一个层负责业务逻辑，另一个层既要负责表现逻辑又要负责控制逻辑[11],这种架构在开发大型项目时，软件的健壮性和可伸缩性都不好。因此在MVC模型中引入了“控制器”的概念，也就是由三个基本部分组成一个应用：这三个模型分别是Model(模型)、View(视图)和Cortoller(控制器)。这三个部分不仅使系统模块间的耦合度比原来更低，又使得系统的各个层次和模块之间可以独立并行开发，从而提高系统的开发效率，缩短系统的开发周期，使得应用的可扩展性及可维护性也得到了提高。但是在Django中，模型(Model)、视图(Views)和模板(Template)在Django中得到了广泛使用，因为控制器会处理从框架本身接受用户输入的部分，这便是MTV模式。

MVC控制器部分由Django框架的URLconf实现。URLconf机制使用正则表达式来匹配URL并调用适当的Python函数。URLconf和URL规则是可选的，没有限制。该框架封装了控制层并与数据进行交互。控制对数据库表的读取，写入，删除和更新操作。编写程序时无需编写大量代码，很多的事情也可以只编写很少的代码就能完成。由于MVC框架功能强大，能完成很多工作，这样一来就提高了工作效率。本系统的设计就用到了djangorestframwork。

## 2.2 相关技术

## 2.2.1 Django技术

（1）用户点击注按钮，将要注册的信息发送给网站服务器。

（2）Controller控制器接收到用户的注册信息后，告知Model层完成在数据库中将用户的注册信息保存的工作。

（3）Model层接到指令后，将用户的注册信息储存到数据库。

（4）保存数据之后将保存的结果返回给Model模型。

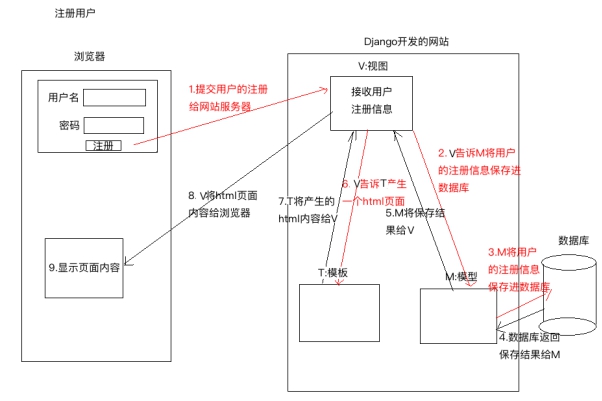
（5）Model层将保存的结果返回给Controller控制器。

（6）保存的结果返回给Controller控制器后，控制器通知View视图，让它产生一个html页面。

（7）View将产生的Html页面的内容给了Controller控制器。

（8）Controller将Html页面的内容返回给浏览器。

（9）浏览器接受到服务器Controller返回的Html页面进行解析展示。



2.1 Django运行图

django的生命周期：

第一步:浏览器发起请求

第二步:由WSGI创建socket服务端，接收请求(Httprequest)

第三步:中间件处理请求

第四步:url路由，依据当前请求的URL来找到视图函数

第五步:view视图,进行业务处理(ORM处理数据，从数据库中提取数据返回给view视图，view视图则将数据渲染到template模板，将数据返回)

第六步:中间件处理响应

第七步:WSGI返回响应(HttpResponse)

第八步:浏览器渲染：浏览器发起请求，django通过wsgi实现一个web服务器接收请求，第一步是中间件处理请求。然后通过url找到相对应的视图函数进行路由分发，返回响应要通过wsgi服务器，最后要通过浏览器渲染出来。

## 2.2.2 ORM技术

ORM（Object-Relational Mapping）称为对象关系映射，它允许在关系数据库和业务实体对象之间创建映射，因此就无需使用复杂的SQL语句对业务对象执行某些操作，所要掌握的就是如何操作对象属性和方法。对数据库的操作使用ORM完成，Django-orm通过映射的方式连接mysql数据库，执行sql语句。

# 第3章 系统需求分析

## 3.1 系统可行性分析

### 3.1.1 系统的必要性

利用城市公交信息查询系统能够准确地进行站点査询、线路査询和换乘方案査询，不仅最大程度上满足了乘客对公交信息的查询需求，而且也提高了城市公交系统的运作效率。随着智能科学技术的不断升级换代，其处理能力逐步增强，涵盖了生活的方方面面。公交信息查询系统即是在时代背景下应运而生的产物，该系统具有查询公交车站点和查询公交车线路以及提供换乘线路等功能。公交车在我国是城市里人民的主要出行的工具，也是人民方便出行和城市绿色的承担者，因此需要建立气一套方便查询公交问情况的的公交查询系统。为此公交查询系统，越来越受到人们的欢迎。

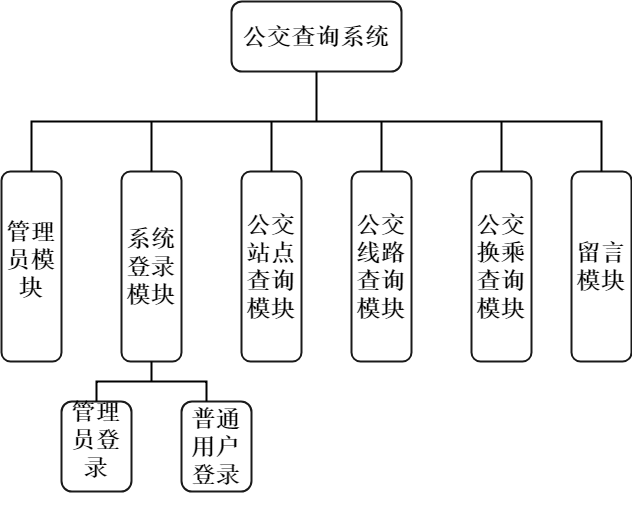
### 3.1.2 系统的技术可行性

在当今这个信息化的时代，互联网已深入世界的各个角落，尤其是移动设备的飞速发展而诞生的基于python编程的开发技术。随着信息化技术大刀阔斧的变革和发展，计算机和信息化在人们的生活中占据了越来越重要的作用，它们的作用不容小觑。在现代科技世界的建设中，互联网技术发挥着举重若轻的作用。计算机技术、多媒体技术、现代通信技术和信息化技术，四位一体，都为促进社会发展和经济增长作出了巨大贡献。

现如今，人口规模的迅速增长和城市规模的急遽扩大，城市交通问题日益涌现。规划好每一条公交线路，不仅要考虑诸多因素，还要站在市民的角度上去思考。公交车是人们日常出行中必不可少的交通工具，然而面对繁芜多杂的公交线路常常让人混淆错认。但是由于当今互联网发展速度快，且每家每户基本都有互联网，因此如今乘坐公交车可以借助互联网来使的乘坐公交更加的便利简单，而且还可以通过公交查询系统来节约个人的时间。公交查询系统采用的是前后端分离的技术，使用的是MVT的框架模型，前端采用html、css、js，后端用的是Django，数据库采用的是Mysql数据库，django-orm通过映射的方式连接mysql数据库，执行sql语句。后端主要有注册、登录、留言、站点查询、线路查询、换乘查询，这几个模块。数据表主要有用户，站点，线路，留言，城市和线路站点多对多关系这6个数据表，在站点表、线路表和城市表都有外键关系，关联着城市表中的id，因此可以通过设定不同城市查找出对应城市的站点信，线路，换乘信息。登录模块是用的JWT认证方式，后端通过restframework的试图类来对数据进行控制，首先通过request查找出数据库的数据，再通过序列化器，把数据进行序列化后，通过response将数据返回，前端调用相对应的api接口即可获取数据，通过js将数据进行可是化显示。

## 3.2 系统需求概述

该系统主要建立一个基于python的公交查询系统。公交查询系统有六个模块，分别是“用户登录”、“公交站点查询”、“公交线路查询”、“换乘查询”和“留言”模块。本系统功能模块如图3.1所示。



图

图3.1系统功能模块图

## 3.3 需求概述

根据市场需要，对公家查询系统系统的设计提出以下要求：

(1)显示的界面应该简单大方，能够突出重点，操作简单。

(2)要求有管理员登陆和用户登录。

(3)给用户提供查询公交站点。

(4)为用户提供公交路线查询的功能。

(5)为用户提供给公交换乘查询的功能。

(6)有留言功能，方便用户反馈信息。

## 3.4 设计原则

1. 实用性原则。让设计的系统可以有效地解决实际生活中的问题，而不是一个

中看不中用的花架子。设计一个系统要看它有无实用性，所以这影响着系统成败。是系统设计中最关键最基本的原则。

1. 可靠性原则。系统的可靠性是必不可少的，它不仅是系统设计的着力点,也

是能否提供给用户一个运行稳定的系统的根本所在。众所周知，一个故障频繁的系统,不仅不能够使用户的正常需求得到满足，还会带来诸多麻烦。

1. 安全性原则。该公交查询系统分为管理员和用户两类，管理员可以拥有一些

权限来提供给系统相应的功能，而用户拥有相应的账户和密码。

1. 简单操作原则。系统的操作简单是设计过程中很值得注意的问题，让精心设

计好的用户接口能够使的系统用户界面友好，以便于用户好操作和理解。

第4章 系统各部分设计和功能

## 4.1 数据库设计

数据库是公交查询系统的根基,数据库的设计质量必须重视。它直接关系到公交查询系统的设计成败与否。本节详细介绍数据库设计。

数据库是用于管理数据的最新技术。近年来数据库管理系统一直在不断发展，并已逐渐演变为实用性和通用性兼并的系统软件。数据库具有许多优点，包括数据结构化，数据冗余度低，易于扩充等这些优点。目前有相当多一部分较为庞大的信息系统的设计都是建立在数据库之上的。不管是在微型机或者是中小型计算机中，甚至是大型计算机中都配置了数据库管理系统。

正是因为数据库系统的问世，本来以加工数据的程序为中心的信息系统迈向了一个新阶段，它的中心转换成了共享的数据库。如此一来不仅使得应用程序的研发和维护更加方便快捷，又对数据的集中管理大有裨益。在使数据的利用性和相容性得到提高的同时，又显现出了决策的可靠性。现如今数据库已经占据了现代信息系统中不可或缺的重要位置。在当今的计算机领域发展速度最快的技术中，数据库技术就占有一席之地。

数据库和它自己应用的性能是基于有良好性能的数据库设计之上的，一个没有得到完善设计的数据库，会很大程度上限制于数据库的性能。进行操作的基础就是数据库中的数据，数据库设计的重点就在于怎么才能使用户的数据得到合理存储，怎么才能提高给用户方便简洁的数据处理方式。

每件事物的存在都要遵循一定的原则，数据库的设计也是如此。在关系数据库中，要遵循的规则就是范式。这是一种关系模式的集合，它符合某一种级别。人们设计数据库时通常会遵循第三范式。即：已经在其他表中包含了的非主关键字信息就不会再包含于数据库表中。范式这个规则的应用不仅降低了数据冗余，节省了存储空间，也让增、删、改的速度变得更快了。

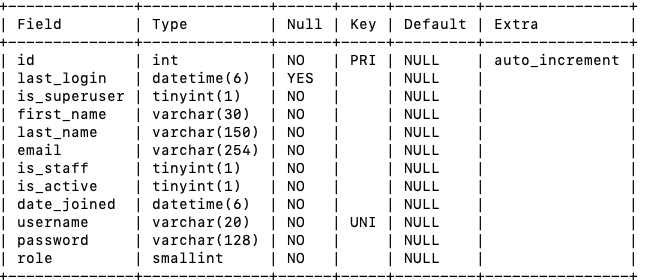
公交查询系统中数据库采用的是Mysql数据库。Django-orm通过映射的方式连接数据库，执行sql语句。整个系统所包含的的数据库信息有车辆信息、公交站信息、公交线路信息、留言内容等。数据表主要有用户，站点，线路，留言，城市和线路站点多对多关系这6个数据表，在站点表、线路表和城市表都有外键关系，关联着城市表中的id，因此可以通过设定不同城市查找出对应城市的站点信，线路，换乘信息

### 4.1.1 数据表的设计

(1)用户信息表

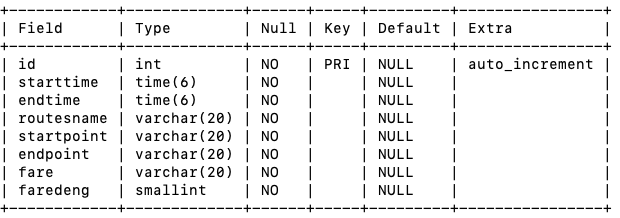
用户信息表的作用是保存用户信息。该数据表的结构如表4.1所示。

表4.1 用户信息表结构

(2)线路信息表

线路信息表用于保存公交线路信息。该数据表的结构如表4.2所示。

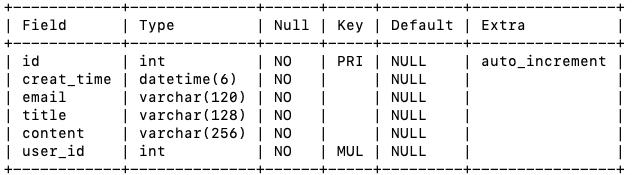
表4.2 站点信息表结构



1. 留言信息表

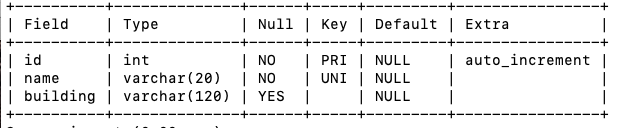
留言信息表用于保存用户的留言信息。该数据表的结构如表4.3所示。

表4.3 留言信息表结构

(4)站点信息表

站点信息表主要用于站点信息的保存。该数据表的结构如表4.4所示

表4.4 站点信息表结



### 4.1.2 数据库功能

整个系统模块可以按照实际应用需求实现各种功能，还可以利于管理员登录系统以及时修改和添加信息。数据库Mysql不仅实现了对公交车站点、公交车车次、公交线路以及用户留言内容数据库的创建，而且还实现了管理员对公交信息的插入、删除、修改和留言版的查看等功能，从而使用户能运用此系统更加方便的查看相关公交信息，也节省了大量的时间等特点。

实现各个功能程序如下：

class UserViewSet(viewsets.ModelViewSet):

queryset = User.objects.all()

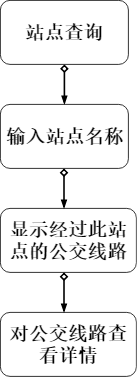
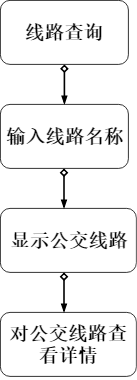
serializer\_class = UserSerializer

permission\_classes = [AdminPermission] #权限设置，只有管理员才能登录

# filterset\_fields = ['is\_active']

### 4.1.3 系统数据查询思路图

公交查询系统中用户可以用三种查询方式查询公交车信息，这些查询方式分别是公交车线路查询、公交车站点查询以及换乘查询;还可以实现用户留言以及用户查看留言;管理员对公交车信息经行操作，其中管理员对公交车信息可以进行增添、删除以及更新;管理员可以在后台审阅用户的留言信息，对用户的留言进行操作。其中站点查询和线路查询具体的系统流程图如图4.5所示。

<a>.站点查询流程图 <b>.线路查询流程图

图4.5 系统查询思路

## 4.2公交查询系统各模块功能

### 4.2.1 系统登录模块

登录模块有两种，分别是管理员登录和用户登录两大部分。不同登录模块需要输入相应的账户和密码，用户选择身份后，对于登录成功的普通用户，登录后进入用户个人界面进行相应操作，对于登录成功的管理员，则进入不同于用户界面的管理员界面，从而对公交车信息进行相应操作。如果输入的账号或者是密码是错误的，则系统会给出错误提示，此时需要重新进行输入。

1. 管理员登录

输入注册的账号和密码后管理员就能成功登录进公交查询系统的管理员界面，才有权限进行各种操作。

1. 普通用户登录

用户登录模块主要是先要求用户注册，再对用户进行登录信息的，如果账号和密码都正确就可以成功登录系统，该模块主要是能实现用户随时随地进查询乘车信息。

### 4.2.3 管理员模块

本模块对路线信息进行增加与删改，编辑线路信息和站点信息。也可在此模块查看普通用户登录的账号和密码与其留言信息。这个功能模块的作用是对乘车有关数据库的创建、车次和车站以及线路的插入、车次和车站以及线路的修改、车次和车站以及线路的删除查询。

(1)乘车相关的数据库的创建：创建一个相关数据库，表名为“busroutes”；

(2)车次和车站以及线路的插入：便于管理员能随时录入新增的乘车信息；

(3)车次和车站以及线路的修改：便于管理员能随时对于作出乘车信息修改；

(4)车次和车站以及线路的删除：便于管理员能对不当的乘车信息进行删除操作；

### 4.2.3 公交站点查询模块

本模块既可以在不用注册和登录而是以游客身份进入公交查询系统的状态下进行查询，也可以在用户成功登录公交查询系统后进行查询，用户根据需要在搜索框中的键入站点名称，就能查询到经过这个站点的所有公交线路。

### 4.2.4 公交线路查询模块

本模块同样不需注册登录便可进行查询，使用者输入想要查询的线路名称，就可以查询到包含在该线路内的按顺序经过的所有站点。

### 4.2.5 换乘查询模块

用户在搜索框中输入起始站和终点站名称，系统就会给出换乘的路线方案，查询到到达目的站点所需要乘坐的公交车次。

### 4.2.6 留言模块

用户成功注册登录后可以在此模块留下自己对公交查询系统的意见与建议，以供公交管理人员根据用户的合理建议改进不足之处，更好的完善系统。

# 第5章 系统的实现

## 5.1 公交查询系统主页

用户可以以游客身份进入公交查询系统，此时可以实现公交查询功能，若要对公交查询系统进行留言或者查看个人信息还需要登录进去公交查询系统。



图5.1 公交查询系统主页

## 5.2 登录注册模块的实现功能

### 5.2.1 登录注册模块概述

（1）登录注册模块

这个模块需要用户先在系统中注册账号，再使用注册的用户名和密码登录进入公交查询系统。由于登录使用的是JWT认证方式，服务器根据用户名和密码是否相匹配来决定用户能否登录系统。用户必须输入正确的用户名和密码才能进入公交查询系统，登录信息有错误时系统会提示错误，并不再允许用户进行下一步操作。用户注册、登录主页面下如图所示。

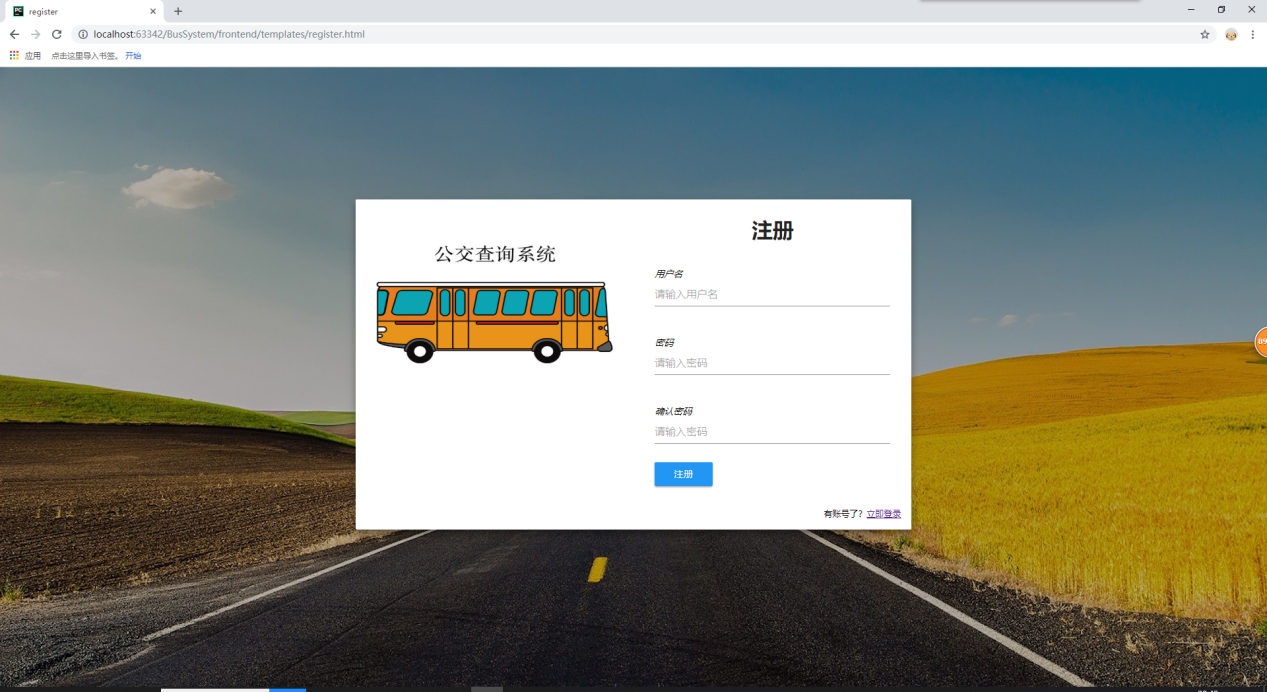


图5.2 用户注册界面

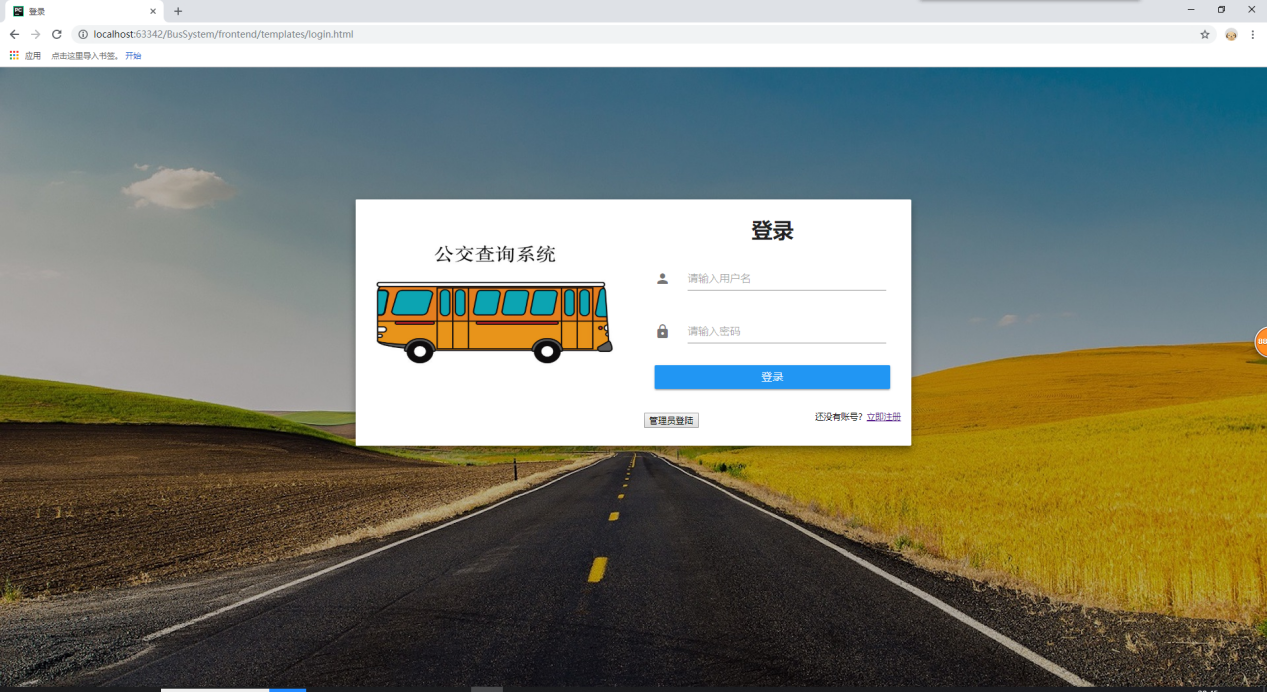


图5.3 用户登录界面

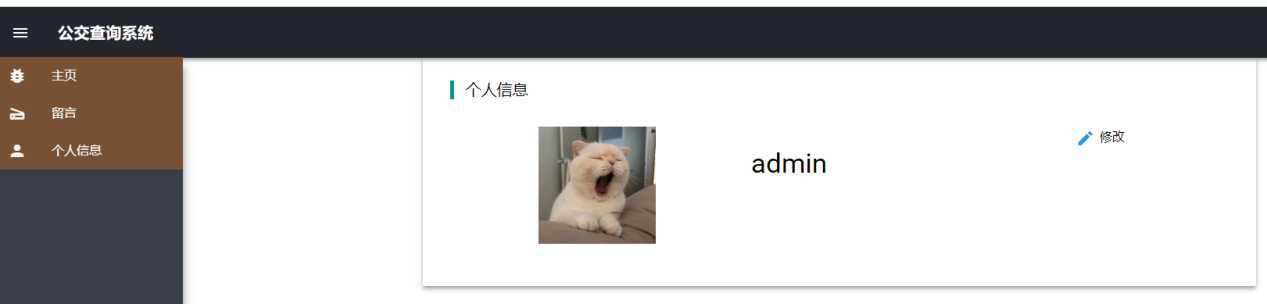


图5.4 用户登录成功后的个人信息

（2）管理员登录模块

管理员登录主要是为了让系统更好的管理，让一个管理员可以及时的插入、删除以及修改车次及站点信息，是广大用户能够及时准确的了解最新的公交车及站点信息，不会使广大用户乘坐错误的公交车耽误用户的时间，使用户能够更舒服的享受本系统带给用户的方便。管理员登陆后来到公交查询系统管理界面，在操作界面中可以对--些用户留言进行回复以及删除。管理员登录及登陆后首页界面如图5.5，图5.6所示。

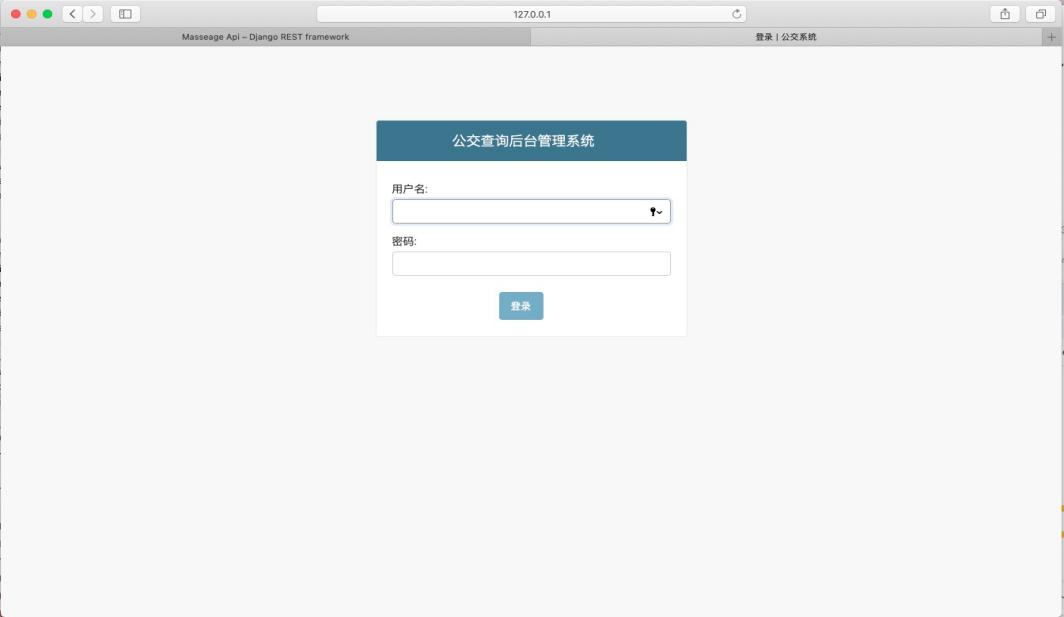


图5.5 管理员登录界面



图

图5.6 管理员首页界面

### 5.1.2 登录模块流程图

（1）用户登录模块

用户进入公交查询系统时先在登录页面填写好用户账号和用户密码，如果用户名及密码和注册时的能够匹配成功，那么就能成功登录并且转到首页界面，否则就显示为错误，则用户需要重新输入输入密码或用户名。程序流程图如图5.3程序流程图所示。

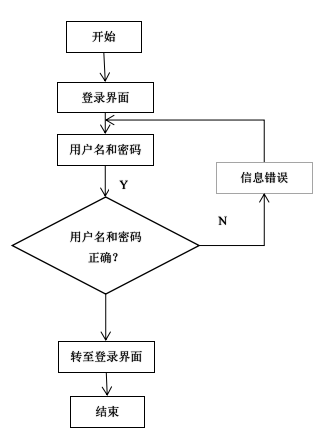


图5.7 用户登录流程图

（2）管理员登录模块

管理员输入账户和密码，接着管理员可以及时的插入、删除以及修改车次及站点信息，以及对一些用户留言进行回复以及删，流程图如下:

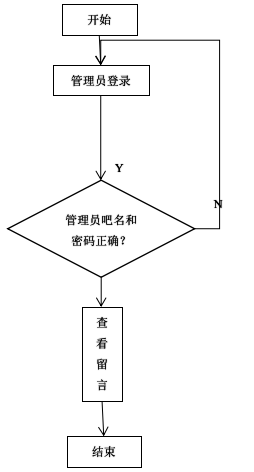


图5.8 留言流程图

管理员对数据进行插入、删除和修改操作：

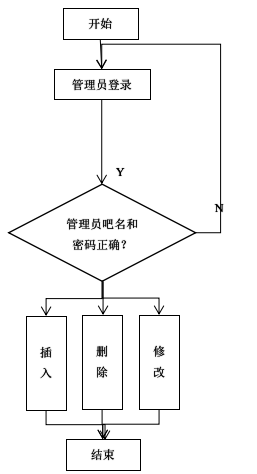


图5.9 管理员操作流程图

### 5.1.3 登录注册模块技术分析

在登录注册模块中，用permission对用户设置权限。只有管理员有对信息进行增加修改和删除的权限，用serializer将数据库里取出的数据进行序列化和反序列化。用户登录时，服务端接收到请求后先验证Token，从Token中获得用户Id，并取得header和playload对头部（header）以及载荷（playload）内容进行签名，生成Signature与之对比匹配则通过。

注册：

class RegisterAPIView(GenericAPIView):  
 serializer\_class = RegisterSerializer  
 permission\_classes = ()  
  
 def post(self, request):  
 serializer = self.get\_serializer(data=request.data)  
 if serializer.is\_valid():  
 serializer.save()  
 return Response({'message': '用户注册成功'}, status=status.HTTP\_200\_OK)  
 else:  
 confirm\_password = serializer.data.get("confirm\_password")  
 password = serializer.data.get("password")  
 username = serializer.data.get('username')  
 if User.objects.filter(username=username).exists():  
 return Response({'message': '用户名已存在'},status=status.HTTP\_401\_UNAUTHORIZED)  
 elif password != confirm\_password:  
 return Response({'message': '两次密码不一样', \*\*serializer.errors}, status=status.HTTP\_400\_BAD\_REQUEST)  
 return Response({'message': '信息填写不完整'}, status=status.HTTP\_400\_BAD\_REQUEST)

登录：

class LoginAPIView(JSONWebTokenAPIView):  
 """  
 用户登陆视图: JWT认证  
 """  
 serializer\_class = JSONWebTokenSerializer  
  
 # todo: 登陆逻辑有优化空间  
 def post(self, request, \*args, \*\*kwargs):  
 serializer = self.get\_serializer(data=request.data)  
 # username = request.data.get('username')  
 # if username:  
 # try:  
 # user = User.objects.get(username=username)  
 # except User.DoesNotExist:  
 # return Response({'message': '用户未注册'}, status=status.HTTP\_400\_BAD\_REQUEST)  
 # if not user.is\_active:  
 # return Response({'message': '当前用户未审批, 请联系管理员进行审批操作'}, status=status.HTTP\_400\_BAD\_REQUEST)  
 if serializer.is\_valid():  
 user = serializer.object.get('user') or request.user  
 token = serializer.object.get('token')  
 response\_data = jwt\_response\_payload\_handler(token, user, request)  
 response = Response(response\_data)  
 if api\_settings.JWT\_AUTH\_COOKIE:  
 expiration = (now() + api\_settings.JWT\_EXPIRATION\_DELTA)  
 response.set\_cookie(api\_settings.JWT\_AUTH\_COOKIE, token, expires=expiration, httponly=True)  
 return response  
 else:  
 errors = serializer.errors  
 if 'non\_field\_errors' in errors:  
 return Response({'message': '用户名或密码错误'}, status=status.HTTP\_400\_BAD\_REQUEST)  
 return Response({'message': '用户信息不能为空'}, status=status.HTTP\_400\_BAD\_REQUEST)

## 5.2 管理员模块设计

### 5.2.1 管理员模块设计概述

管理员进入到公交查询系统的管理员界面时，便可以对相应的功能进行选择，可以对公交线路、公交站点进行增加和删改。还可以查看登录公交查询系统的用户和密码，查看用户的留言信息。例如要添加一条公交线路，首先添加路线信息，再添加站点信息，最后进行路线管理，将站点添加到公交路线中去。如图5.10,图5.11，图5.13，图5.14所示。

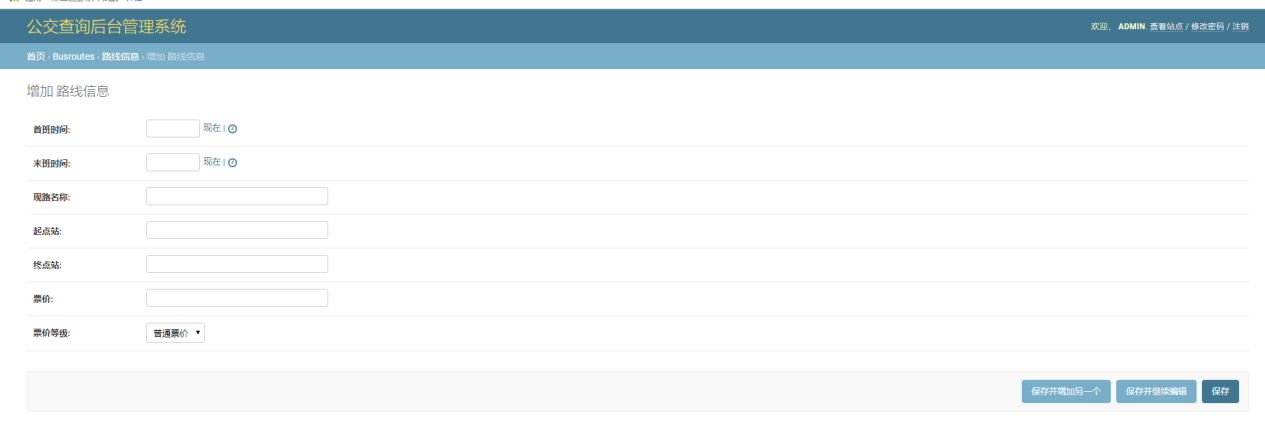


图5.10 增加路线信息

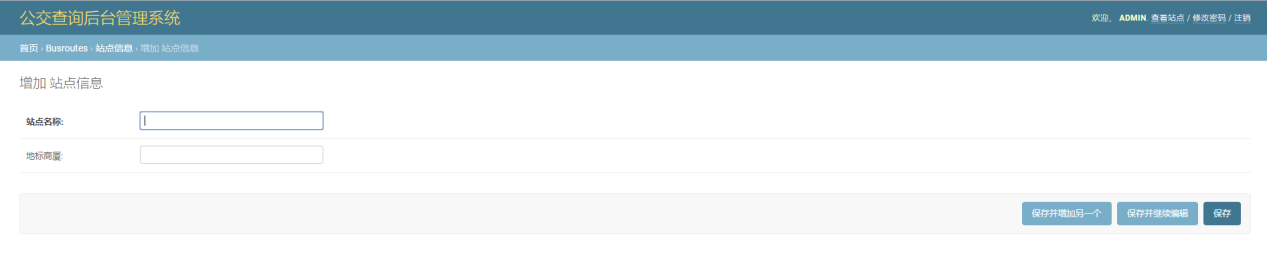


图5.11 增加站点信息

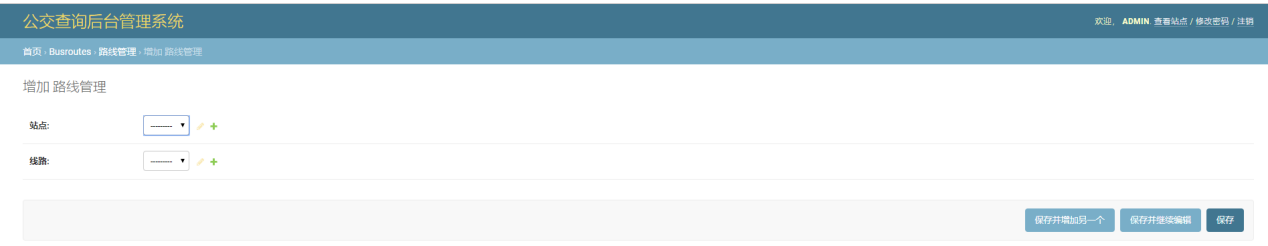


图5.12 增加线路管理

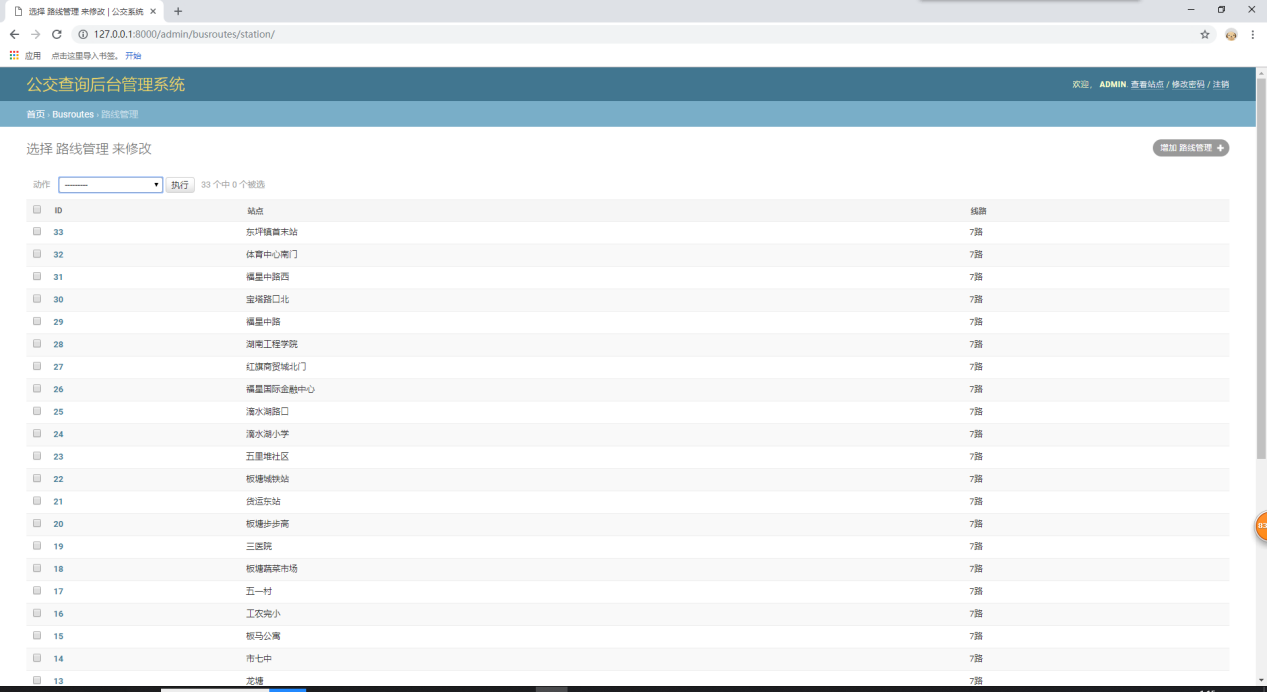


图5.13 进行路线管理



图5.14 查看用户留言

### 5.2.2 管理员模块流程图

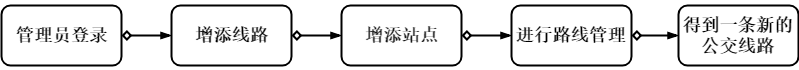


图5.15 管理员增添线路操作流程图

### 5.2.1 管理员模块技术分析

在本模块中，从model中获取管理员相关数据，利用Django技术来实现诸个功能。在“站点信息”、“路线信息”、“路线管理”上各绑定了一个点击事件监听器，当管理员用户需要对线路进行相应操作时，可以通过点击“路线信息”相应按钮来查询来进行相应功能的实现。创建数据库按钮是用来实现管理员创建一个线路管理数据库；站点添加按钮则实现的对新站点的录入，删除按钮则实现对于线路或站点的删除功能。

管理员登录代码：

from django.contrib import admin  
from .models import \*  
  
class UserAdmin(admin.ModelAdmin):  
 list\_display = ['id','username','password','role']  
 search\_fields = ["username"]  
class LeaveMessageAdmin(admin.ModelAdmin):  
 list\_display = ['id', 'user', 'creat\_time', 'email','title','content']  
 # search\_fields = ["username"]  
  
admin.site.register(User,UserAdmin)  
admin.site.register(LeaveMessage,LeaveMessageAdmin)

对公交站点和公交线路进行操作的代码：

class BusRoutesAdmin(admin.ModelAdmin):  
 list\_display = ['id','routesname','startpoint','endpoint','fare','faredeng','starttime','endtime']  
 search\_fields = ["routesname"]  
  
class BusStationAdmin(admin.ModelAdmin):  
 list\_display = ['id', 'name', 'building']  
 search\_fields = ["name"]

class StationAdmin(admin.ModelAdmin):  
 list\_display = ['id', 'station', 'routers']  
 # search\_fields = ["name"]  
admin.site.register(BusRoutes,BusRoutesAdmin)  
admin.site.register(BusStation,BusStationAdmin)  
admin.site.register(Station,StationAdmin)

class BaseModel(models.Model):  
 starttime = models.TimeField(verbose\_name="首班时间")  
 endtime = models.TimeField(verbose\_name="末班时间")  
  
 class Meta:  
 abstract = True #不创建表  
  
#线路表  
class BusRoutes(BaseModel):  
 FareChoices = (  
 (0, '普通票价'),  
 (1, '特殊票价'),  
 )  
 routesname = models.CharField(max\_length=20,verbose\_name="现路名称")  
 startpoint = models.CharField(max\_length=20,verbose\_name="起点站")  
 endpoint = models.CharField(max\_length=20,verbose\_name="终点站")  
 fare = models.CharField(max\_length=20,verbose\_name="票价")  
 faredeng =models.SmallIntegerField(choices=FareChoices,default=0,verbose\_name="票价等级")  
  
 class Meta:  
 db\_table = 'busroutes'  
 verbose\_name ="路线信息"  
 verbose\_name\_plural = verbose\_name  
  
 def \_\_str\_\_(self):  
 return "%s"%self.routesname  
  
#站点表  
class BusStation(models.Model):  
 name = models.CharField(max\_length=20,unique=True,verbose\_name="站点名称")  
 building = models.CharField(max\_length=120,null=True,blank=True,verbose\_name="地标商厦")  
  
 class Meta:  
 db\_table='busstations'  
 verbose\_name = "站点信息"  
 verbose\_name\_plural = verbose\_name  
  
 def \_\_str\_\_(self):  
 return "%s"%self.name  
  
#关联表  
class Station(models.Model):  
 station = models.ForeignKey(BusStation,related\_name='station\_info',on\_delete=models.CASCADE,verbose\_name="站点")  
 routers = models.ForeignKey(BusRoutes,related\_name='routes\_info',on\_delete=models.CASCADE,verbose\_name="线路")  
  
 class Meta:  
 db\_table = 'station'  
 verbose\_name = "路线管理"  
 verbose\_name\_plural = verbose\_name

## 5.3 公交站点查询模块的实现

### 5.3.1 公交站点查询模块概述

公交车站点查询模块是在用户进入系统后，在搜索框内搜索相关的站点的全名，此时会按站点的确切信息匹配查询，就会得到相关站的的公交车次以及有关此站点的公交路线，如果数据库没有相关的站点，那么就不能得到相关的信息，这就是完全查询。另外，如果不知道站点的全名，也可以只输入站点名称和中的部分字符，系统会给出与输入的字符相关的站点信息模糊查询。例如输入一个确切站点的名称，如板塘城铁站后，得到的公交线路信息如图5.16所示：



图5.16 公交站点查询

### 5.3.2 公交站点查询模块技术分析

公交站点查询模块中用到了orm技术对数据库进行操作，查找出数据然后使用serilazers序列化器进行序列化。

class SiteSearchSrializer(serializers.ModelSerializer):  
 """  
 通过站点查出几条公交线路经过此地,并且返回站点信息  
 """  
 station\_info = StationSerializer(many=True,read\_only=True)  
 class Meta:  
 model = BusStation  
 fields = [  
 'id',  
 'name',  
 'building',  
 'station\_info'  
 ]

class SiteSearchViewSet(viewsets.ModelViewSet):  
 """  
 #通过站点查出几条公交线路经过此地,并且返回站点信息  
 """  
 permission\_classes = []  
 queryset = BusStation.objects.all()  
 serializer\_class = SiteSearchSrializer  
 search\_fields = ['name','building']

## 5.4 公交线路查询模块的实现

### 5.4.1 公交线路查询模块概述

公交线路查询模块的实现步骤是在用户进入系统后，在搜索框内搜索想查询的公交线路，经过查询后就会得到相关的公交车次，同时也会显示途径此公交线路的所有公交站点，如果数据库中没有要查找的线路，就会报错。例如要查询7路公交时在搜索框内搜索7路公交车，就会出现此次公交车及其详细信息，如图5.17所示。



图5.17 公交线路查询

### 5.4.2 公交线路查询模块技术分析

公交线路查询模块的实现方法与公交站点查询类似。

class SiteSrializer(serializers.ModelSerializer):  
 class Meta:  
 model = BusStation  
 fields = "\_\_all\_\_"  
  
class RoutersSerilazer(serializers.ModelSerializer):  
 class Meta:  
 model = BusRoutes  
 fields = "\_\_all\_\_"  
  
class StationSerializer(serializers.ModelSerializer):  
 routers = RoutersSerilazer(read\_only=True)  
 class Meta:  
 model = Station  
 fields = "\_\_all\_\_"  
  
class StationRoutersSerializer(serializers.ModelSerializer):  
 station = SiteSrializer(read\_only=True)  
 class Meta:  
 model = Station  
 fields = "\_\_all\_\_"

class RoutersSearchSerilazer(serializers.ModelSerializer):  
 routes\_info = StationRoutersSerializer(many=True,read\_only=True)  
 faredengs = serializers.CharField(source='get\_faredeng\_display',read\_only=True)  
 class Meta:  
 model = BusRoutes  
 fields = ['id','routesname','startpoint','endpoint','fare','faredengs','routes\_info']

class BusRoutesViewSet(viewsets.ModelViewSet):  
 *"""* 搜索车次并且查看车次的站点 *"""* permission\_classes = []  
 queryset = BusRoutes.objects.all()  
 serializer\_class =RoutersSearchSerilazer  
 search\_fields = ['routesname']  
  
class BusRoutesInfoVieSet(GenericAPIView):

## 5.5 公交换乘查询模块的实现

### 5.5.1 公交换乘查询模块概述

用户登录进入系统后在搜索框中输入起始地和目的地，系统在数据库中查找相关信息，最后给出符合要求的公交线路及其详细信息，提供换乘方案，如图5.18所示。



图5.18 公交换乘查询

### 5.5.2 公交车换乘查询模块技术分析

公交换乘查询模块的实现中也用到了orm技术对数据库进行操作，查找出数据然后使用serilazers序列化器进行序列化。公交换乘查询模块中需要用到数据库中的站点表，线路表以及线路关系表，第一步是先根据用户要查询的起始站点名和终点站点名获取经过这两个站点的线路名，然后进行查询。搜寻数据库中包含这段起始站点和终点站的公交线路，找出它们的交集站点，得到换乘站。意思就是途径起始站可以到达的所有站点形成一个集合甲，途径终点站可以到达的所有站点形成另一个集合乙，那么换乘就是在求集合甲与集合乙之间有没有可以连通的线路。这样就找到了作为中间的换乘线路。

import json, requests, time

class Huancheng:

def \_\_init\_\_(self):

self.url = "https://api.jisuapi.com/transit/station2s"

def request\_d(self,start,end,city):

data = {}

data["appkey"] = "3a4a6ac95a6b3669"

data["start"] = start # 湘潭

data["end"] = end # utf8

data["city"] = city

data["type"] = ""

respone = requests.get(url=self.url,params=data).content.decode()

return respone

def extract(self,res)

jsonarr = json.loads(res)

return jsonarr

def run(self,stat,end,city):

response = self.request\_d(stat,end,city)

date = self.extract(response)

return date

class BusRoutesInfoVieSet(views.APIView):

公交换乘:{

1、直线叫站点

2、一次换乘

3、二次换乘

}

permission\_classes = []

def get(self, request, \*args, \*\*kwargs):

starting\_point = self.request.query\_params.get('starting\_point') # 起始地

destination = self.request.query\_params.get('destination') # 目的地

city = self.request.query\_params.get('city')

Transfer\_data = Huancheng().run(starting\_point,destination,city)

print(Transfer\_data)

if Transfer\_data.get("status") == 0:

return Response(Transfer\_data.get("result"),status=status.HTTP\_200\_OK)

else:

return Response({'code':Transfer\_data.get("status"),'msg':Transfer\_data.get("msg")},status=status.HTTP\_500\_INTERNAL\_SERVER\_ERROR)

## 5.6 留言模块的实现

### 5.6.1 留言模块概述

用户成功登录进公交系统后，可根据自己对此系统的缺点和不足之处提出自己的意见或者是建议并进行留言。管理员成功登录后可对用户的留言进行查看，如图5.19，图5.20所示。

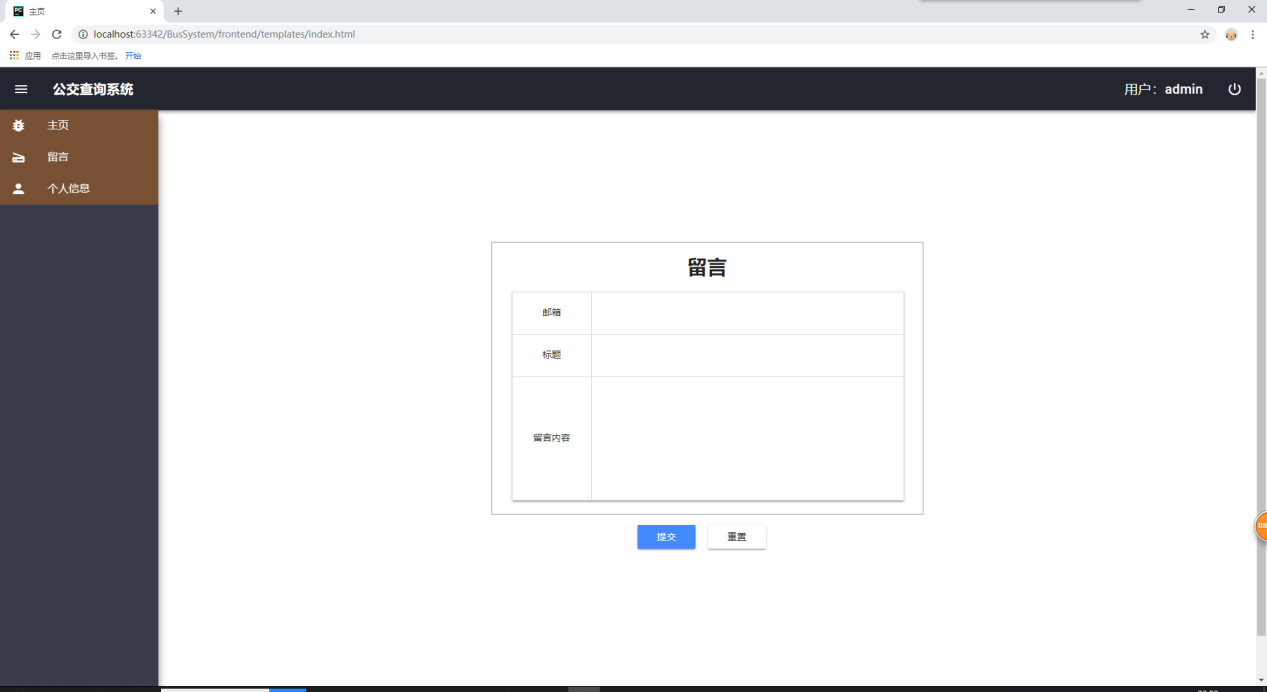


图5.19 用户留言界面



图5.20 查看登录的用户信息

### 5.6.2 留言模块技术分析

留言模块的实现需要创建一个数据表来存储用户在留言时留下的个人信息，于是创建一个留言信息表，管理员查看留言信息时从留言信息表中调取数据。以下为用户留言代码：

class LeaveMessage(models.Model):  
 user = models.ForeignKey(User, related\_name='user',on\_delete=models.CASCADE,verbose\_name="用户")  
 creat\_time = models.DateTimeField(auto\_now\_add=True,verbose\_name="留言时间")  
 email = models.CharField(max\_length=120,verbose\_name="邮箱")  
 title = models.CharField(max\_length=128,verbose\_name="留言标题")  
 content = models.CharField(max\_length=256,verbose\_name="留言内容")  
  
  
 class Meta:  
 db\_table = 'leavemessage'  
 verbose\_name = "用户留言"  
 verbose\_name\_plural = verbose\_name

# 第6章 系统的测试与维护

## 6.1 测试的目的及意义

之所以要对系统进行测试，是要找出发现系统的瑕疵以及错误，进行错误更正和功能完善。在系统的开发过程中往往会由于许多原因而导致系统出现问题，并且不易被发现，这时就需要对系统进行测试，使功能更加全面，使系统更加完善。

另外，测试不仅仅是为了发现系统的错误，还是为了进一步提升系统的使用是否容易操作，要争取做到令用户满意，并能给用户带来方便和帮助。在测试的过程也是用来提升自己的过程，通过系统测试测不仅可以完善系统，还可以检查自己的不足，以此来提升自己的开发能力。

## 6.2 系统部分测试结果

在进行系统测试时，应对系统的各个模块进行测试：

（1）公交车路线查询。用户在登录进入系统后就可以进行公交路线查询，在查询框内输入所乘坐的公交，就能得到该公交的线路，此模块功能正常。如:在登陆界面界面的查询框内搜索“7路”，就可以显示出7路公交的路线及其所途径的所有站点。

（2）公交站点查询。用户在登录进入系统后就可以进行公交车站点查询，在查询框内搜索某个站点，就可以得到哪些公交到此站点，此模块功能正常。如:在查询框内搜索“湖南工程学院”就可以得到到此站的全部公交车信息。

（3）换乘查询。用户在登录进入系统后就可以进行公交车站点查询，在查询框内搜索某个站点，就能得到到此站点的公交规划和换乘信息，此模块功能正常。如：在查询框内搜索“火车站”就可以得到到火车站的公交规划，以及换乘的站点。

（4）公交站点的增添与删改，管理员进入管理界面中的站点信息，对站点进行修

改。此模块功能正常，如图6.1，图6.2所示。

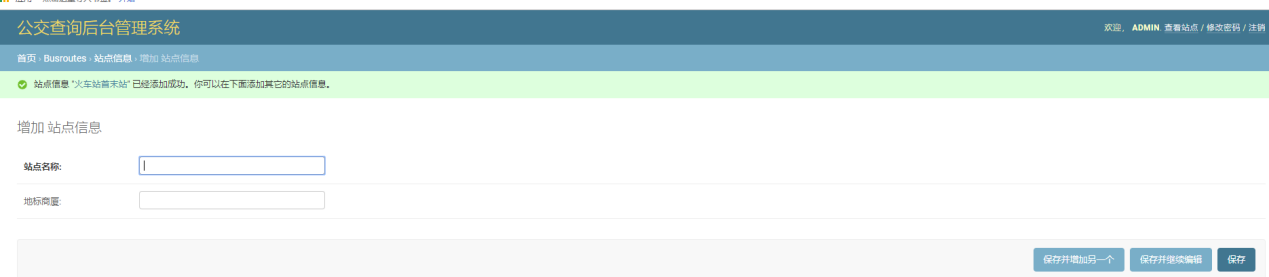


图6.1 增添公交站点成功

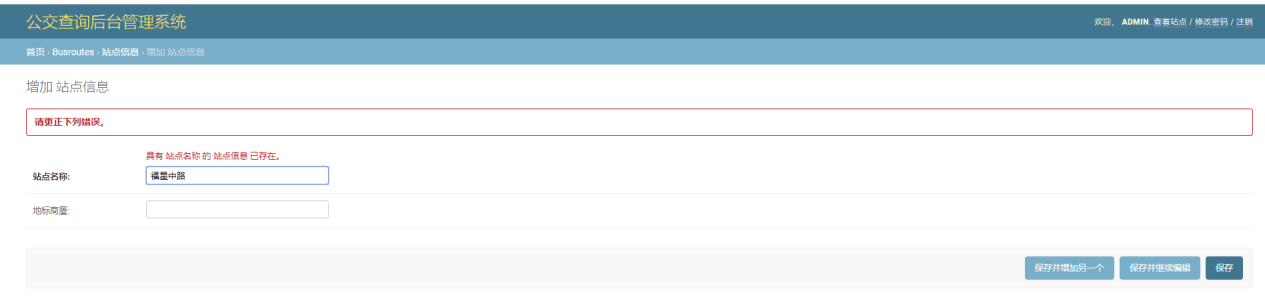


图6.2增添重复的公交站点，系统报错

1. 公交线路的增添与删改：管理员进入管理界面中的站点信息，对站点进行修

改。此模块功能正常，如图6.3所示。



图6.3 增加线路成功

1. 用户的登录与退出，如图6.4，图6.5，图6.6，图6.7所示

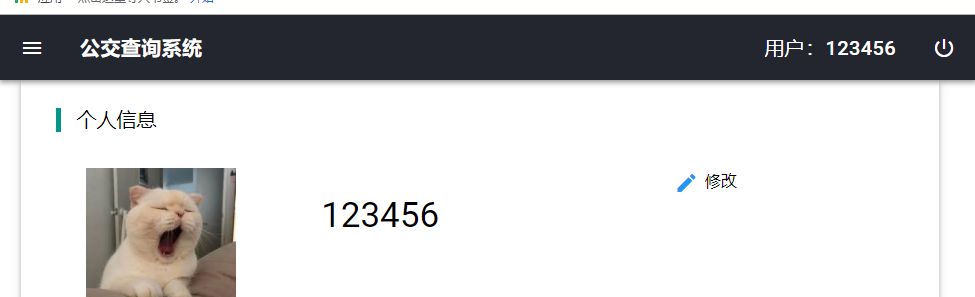


图6.4 成功登录

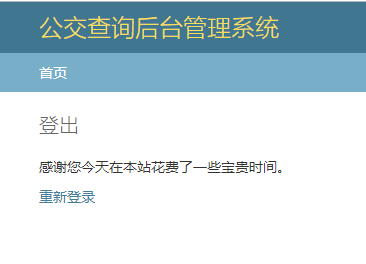


图6.5 成功退出

登录失败：由于用户名或密码错误而导致了登录失败

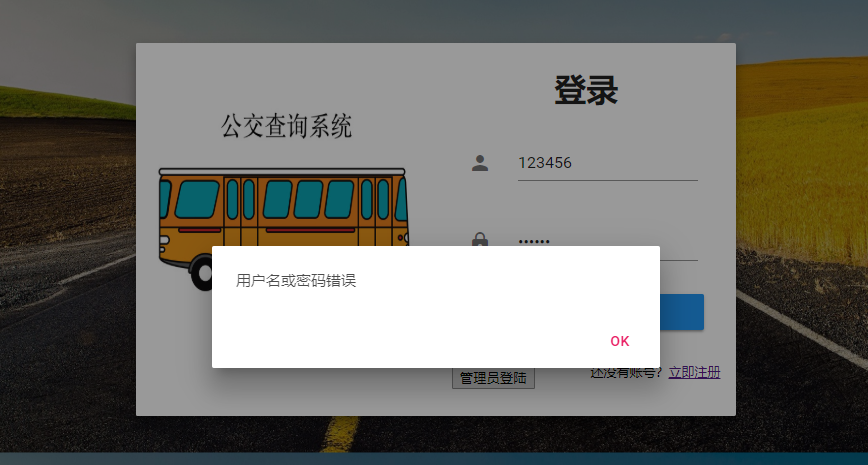


图6.6 登录失败

注册失败：由于设置的密码不符合系统要求导致了注册失败

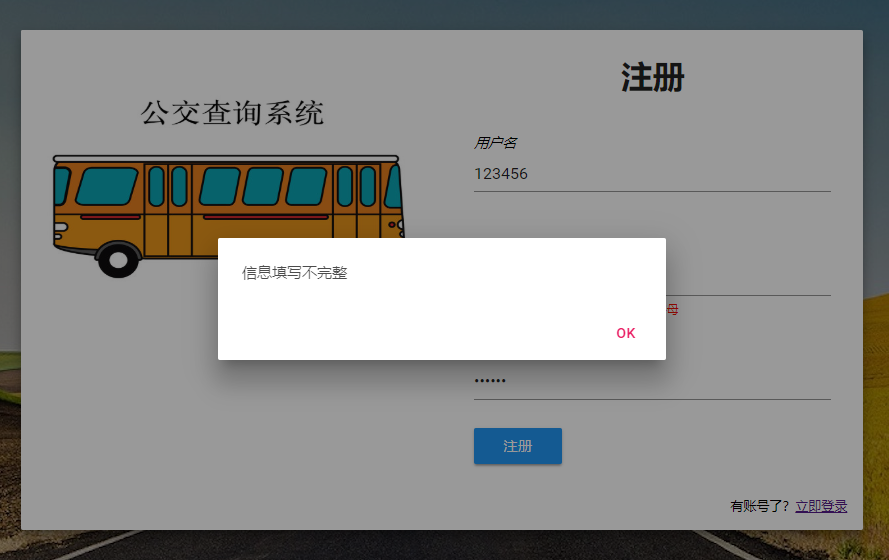


图6.7 注册失败

## 6.3 系统的维护

系统维护是指在系统运行时或维护阶段对对系统所进行的修改。系统的维护是相当耗费耗时的活动。根据不同的系统维护的原因，可以将维护分成三种类型:修改性维护，适用性维护以及完善性维护。

维护的类型不仅只有上述的三种，还有一种被称为预防性维护的维护活动。一般来说是基于要对系统进行改进，或者应对不同需求的目的下才对系统进行维护。换言之，使用技术先进的软件工程方法时，要重新进行设计、编制和测试那些需要维护的软件或者是软件中的某一部分，达到提高软件的可维护性和可靠性的目的。以此来满足用户的需求，也为以后进一步改进系统打下良好基础。

### 6.3.1 影响系统维护的因素

为了维护系统的性能，修正错误和提高实用性，需要注意以下六个方面。

(1)系统的规模大小。

(2)数据库技术的应用。

(3)系统使用寿命。

(4)完成系统所使用的程序设计语言。

(5)所使用的先进软件开发技术。

(6)其他一些因素，如应用的类型、算法的时间复杂度与空间复杂度、任务的难度等。

### 6.3.2 提高系统维护的方法

(1)提前做好需求分析，建立健全明确的优先级目标，保证软件质量。

(2)使用功能更强大的应用技术与相应的工具来提高软件质量。

(3)保证系统的可维护性，进行明确的质量检验与审查。

(4)使用方便维护的程序设计语言。

(5)进一步完善程序文档，改进系统功能。

# 总结

为期几个月的毕业设计已经接近尾声，这几个月的心里历程难以表述，还记得刚要确定课题方向时的茫然无措，之前没有写过论文，不知道要从哪里还是下手，只能四处去查阅资料。我的课题是公交查询系统的设计与实现。这个课题的由来是因为在一次乘坐公交车出行时不知道该乘坐哪一趟车，于是就萌生了要写一个能查询公交线路的系统的想法。确定了课题后就要开始写开题报告了，确定好系统要实现的功能和需要用到的关键技术，我选择用python编程语言来完成公家查询系统的撰写，此前并没有学习过python这门可能，但想给自己一个尝试和挑战。在毕业设计过程中，我一边学习一边完成毕业论文。在网上学习python课程和数据库的使用。这个过程并不容易，有许多知识盲区让我感到挫败又烦闷。比如添加索引不当导致公交站点的顺序出错，环境搭建出了问题，python和mysql语法的不够熟悉，还有其中前端耗费了很多时间，因为先前对这个知识没有掌握，网上的课程和资料也都凌乱无章，很难选择哪些是有用的知识点等等，又因为疫情原因不能返现，在家的创作环境不如在学校的氛围好，而且在家不能像在学校那样能获取更多更详细的资料。这些都是完成毕业设计过程中的拦路虎，给我带来了许多困扰和苦闷。在调整好心情后还是要继续完成毕业设计，从一开始的无从下手，到经过查阅大量资料和文献后对毕业设计有一个大致的思路，再到具体模块框架的设计，这些都要耗费巨大的精力与心思。但我相信勤能补，通过请教老师和询问同学，逐步击破一个又一个问题。

这次的毕业设计让我明白平时专业知识的学习和积累是多么重要，不仅要把握对新知识的学习同时也要注意温习以前学过的知识。另外在不起眼的小细节上也千万不能掉以轻心，微观结构构成宏观设计，只有抱着认真谨慎的态度去处理问题，问题才能够迎刃而解。在对毕业设计的进程上也要把控好时间，平时不赶进度的话就只能在截止日期时手忙脚乱。在完成毕业论文的几个月时间里，我收获了很多也学习到了许多，论文的写作需要不断地进行修改与完善，一边要去研究相关的文献资料遴选出有用的信息，一边要善于总结遇到的困难和它的解决办法，积累经验。还要拓宽自己的思路和知识面，针对公交系统的设计要联系理论和实际，落实到现实生活中在乘坐公交时的需要。这次的毕业设计让我很深刻地明白，不能对不起眼的细节掉以轻心，许多次的报错问题都出在一些很基础的知识上。经过一次又一次的调试，不断发现错误又不断改正错误，这是一项需要细心与耐心的工作。只有克服内心的烦闷苦恼和焦躁，才能迈出下一步。在今后的日子里，不能放松对知识的求索，杜绝眼高手低而要脚踏实地勤勉学习，努力提高自己的专业技能和综合素养。还要勇于直面自己的缺点和不足，认真严谨地对待每一件事。