

本科生毕业论文(设计)

****

**题 目 共享停车平台**

学生姓名 王维天

学 号 201883290467

学 院 计算机与软件学院

专 业 计算机科学与技术

指导教师 马瑞

**二Ｏ二Ｏ 年 五 月 十 日**

目 录

[1 绪论 1](#_Toc41220449)

[1.1 课题研究目的及意义 1](#_Toc41220450)

[1.2 课题国内外研究现状 1](#_Toc41220451)

[1.3 课题研究内容 2](#_Toc41220452)

[1.4 论文组织结构 2](#_Toc41220453)

[2 技术及工具介绍 2](#_Toc41220454)

[2.1 获取用户IP及定位服务 2](#_Toc41220455)

[2.2 IntelliJ IDEA集成开发工具+SSM框架 3](#_Toc41220456)

[2.3 MySQL数据库 3](#_Toc41220457)

[2.4 ECharts介绍 3](#_Toc41220458)

[3 需求分析以及系统环境 3](#_Toc41220459)

[3.1 系统功能需求分析 3](#_Toc41220460)

[3.2 可行性分析 4](#_Toc41220461)

[3.3 系统环境 5](#_Toc41220462)

[4 系统总体设计 5](#_Toc41220463)

[4.1 系统功能结构设计 5](#_Toc41220464)

[4.2 系统功能流程设计 7](#_Toc41220465)

[4.2.1 用户注册登录流程设计 7](#_Toc41220466)

[4.2.2 主页课程推荐流程设计 7](#_Toc41220467)

[4.2.3 课程搜索流程设计 8](#_Toc41220468)

[4.2.4 查看课程流程设计 8](#_Toc41220469)

[4.2.5 观看课程流程设计 9](#_Toc41220470)

[4.2.6 收藏流程设计 9](#_Toc41220471)

[4.2.7 评论管理流程设计 10](#_Toc41220472)

[4.2.8 账户设置流程设计 10](#_Toc41220473)

[4.2.9 反馈功能流程设计 11](#_Toc41220474)

[4.2.10 课程管理功能流程设计 11](#_Toc41220475)

[4.2.11 用户评论管理功能流程设计 12](#_Toc41220476)

[4.2.12 账户信息变更流程设计 12](#_Toc41220477)

[4.2.13 首页轮播管理流程设计 12](#_Toc41220478)

[4.2.14 用户管理流程设计 13](#_Toc41220479)

[4.2.15 机构审批流程设计 14](#_Toc41220480)

[4.2.16 反馈处理流程设计 14](#_Toc41220481)

[4.3 数据库设计 15](#_Toc41220482)

[4.3.1 数据库概念设计 15](#_Toc41220483)

[4.3.2 数据库逻辑设计 18](#_Toc41220484)

[5 系统详细设计 22](#_Toc41220485)

[5.1 系统架构设计 22](#_Toc41220486)

[5.1.1系统前端架构设计 22](#_Toc41220487)

[5.1.2 系统后端SSM架构设计 23](#_Toc41220488)

[5.2 界面设计 25](#_Toc41220489)

[5.3 功能模块详细设计 27](#_Toc41220490)

[5.3.1 用户注册登录详细设计 27](#_Toc41220491)

[5.3.3 搜索功能详细设计 27](#_Toc41220492)

[5.3.4 查看课程功能模块详细设计 27](#_Toc41220493)

[5.3.5 观看网课模块详细设计 28](#_Toc41220494)

[5.3.6 收藏模块详细设计 28](#_Toc41220495)

[5.3.7 评论功能模块详细设计 28](#_Toc41220496)

[5.3.8 账户设置功能模块详细设计 28](#_Toc41220497)

[5.3.9 反馈模块详细设计 29](#_Toc41220498)

[5.3.10 课程管理功能模块详细设计 29](#_Toc41220499)

[5.3.11 用户评论管理功能模块详细设计 29](#_Toc41220500)

[5.3.12 账户信息变更功能模块详细设计 29](#_Toc41220501)

[5.3.13 机构数据中心模块详细设计 30](#_Toc41220502)

[5.3.14 首页轮播管理模块详细设计 30](#_Toc41220503)

[5.3.15 用户管理模块详细设计 30](#_Toc41220504)

[5.3.16 事务审批模块详细设计 30](#_Toc41220505)

[5.3.17 反馈处理模块详细设计 30](#_Toc41220506)

[5.3.18 后台数据中心模块详细设计 30](#_Toc41220507)

[6 系统实现 30](#_Toc41220508)

[6.1 注册登录功能实现 30](#_Toc41220509)

[6.2 主页课程推荐功能实现 31](#_Toc41220510)

[6.3 课程搜索功能实现 33](#_Toc41220511)

[6.4 查看课程功能实现 35](#_Toc41220512)

[6.5 观看网课功能实现 36](#_Toc41220513)

[6.6 收藏功能实现 37](#_Toc41220514)

[6.7 评论功能实现 38](#_Toc41220515)

[6.8 账户设置功能实现 39](#_Toc41220516)

[6.9 反馈功能实现 40](#_Toc41220517)

[6.10 评论功能实现 41](#_Toc41220518)

[6.11 用户评价管理功能实现 43](#_Toc41220519)

[6.12 账户信息变更功能实现 43](#_Toc41220520)

[6.13 机构数据中心功能实现 44](#_Toc41220521)

[6.14 首页轮播管理功能实现 45](#_Toc41220522)

[6.15 用户管理功能实现 46](#_Toc41220523)

[6.16 事务审批功能实现 47](#_Toc41220524)

[6.17反馈处理功能实现 48](#_Toc41220525)

[6.18 后台数据中心功能实现 49](#_Toc41220526)

[6.19 系统部署 51](#_Toc41220527)

[7 系统测试 51](#_Toc41220528)

[7.1 功能测试 52](#_Toc41220529)

[7.2 易用性测试 53](#_Toc41220530)

[8 总结 54](#_Toc41220531)

[参考文献： 54](#_Toc41220532)

[致谢 56](#_Toc41220533)

共享停车平台

王维天

南京信息工程大学计算机与软件学院，江苏 南京 210044

摘要：作为机动车使用者，我们在日常生活中经常会遇到两种矛盾的情况，有时我们停车时找许久都找不到空闲停车位；与此同时，因为车位拥有者早上外出上班，晚上下班回家，白天的居民区车位和晚上的办公楼停车场却利用率非常低，造成停车资源浪费。因此本系统设计提出并实现了一个共享停车使用以及管理平台，希望将不同停车场、个人车位统一管理对用户开放使用，从而缓解车位资源因为时间空间分布不协调造成的资源浪费。本平台系统采用了网页程序的形式，前端使用Vue JavaScript框架技术，后端使用了Python Flask框架编写应用，并以MySQL作为数据库。在部署方面，前端页面使用Nginx作为服务器软件，支持高并发以及负载平衡。通过Docker将前后端打包成镜像，可以在符合条件的服务器上快速部署。共享停车平台允许用户通过名称查询停车位、通过目的地地址查询附近停车位、预定车位、取消订单和订单支付等功能；车位拥有者具有发布车位，管理订单和查看收益和使用率等功能。系统减少了停车资源浪费，方便了车主们出行快速找到车位，也让车位拥有者通过闲置车位收益。系统的用户界面简洁美观，直观易用，可靠稳定。

关键词：共享经济；停车；Vue；JavaScript；Python；Flask；MySQL；Docker

**Shared Parking Platform**

Weitian Wang

School of Computer and Software, NUIST, Nanjing 210044, China

**Abstract:** As car owners, we constantly run into the following dilemma in our daily life––it’s hard to find a parking spot; meanwhile, a large number of parking spots in residential areas and commercial areas are unoccupied during days and nights respectively, since the owners go to work at day and left office at night. This leads to severe underutilization of parking spots. Therefore I propose and implement a shared parking platform that exposes property owned or personally owned parking spots to users for easy usage, in the hope to alleviate the shortage of parking spots caused by uneven resource distribution over time and space. This platform takes the form of a web application, with Vue and JavaScript as its frontend framework, Python Flask library as its backend framework, and MySQL as its database. This project uses Nginx as web server software which enables high concurrency and load balancing of HTTP traffic, For faster and easier deployment across qualified servers, I containerized both frontend and backend as docker images. This shared parking platform enables users to search spots by name, search spots near destination, make reservations, cancel reservations, pay for orders, etc. Spot owners can release parking spots, manage reservations, monitor usage and profit overview, etc. This platform reduces the waste of parking place resources, helps drivers locate spots, and expands owners’ portfolios by leasing their underutilized parking spots. The user interface of this robust shared parking platform is user-friendly and straightforward.

**Key words:** Shared Economy; Parking; Vue; JavaScript; Python; Flask; MySQL; Docker

# 1 绪论

1.1 课题研究目的及意义

我国经济发展上行，人民群众的经济水平不断提高，越来越多的家庭拥有了私家车。分析交通部的数据可以得出，我国机动车现状是保有量大且增速快。截至2021年9月，我国机动车总量达到了3.90亿辆[1]。然而，全国车位预估缺口预估达到8000万个[2]，而且车位数增长缓慢。此外，因为车位在空间上的规划不合理，以及时间上的低周转率，进一步恶化了车位紧张问题。反映到日常生活中，我们经常会遇到找车位难和车位利用率低的矛盾情况和窘境。为了方便车主在日常出行中快速寻找车位，同时帮助物业公司提高车位利用率以及收益，共享停车平台是一个合理的解决方案。传统停车管理解决方案存在以下问题：

1. 不同停车场使用不同的管理计费系统，车主无法充分利用目的地附近所有停车场的资源（空间上利用率不平衡）。
2. 现有系统无法将具体管理到单个停车位，无法知道具体车位在特定时段是否会被占用（时间上利用率不可控）。因此，围绕共享经济的概念，建立统一的停车管理平台才能解决现有问题。

实现一个功能强大且强大易用的共享停车平台，并对不同物业公司的停车场使用统一的软件系统管理，有助于车主们在日常出行中更快找到车位，物业提升闲时车位利用率以及停车费收益。从长远看，减少车辆在停车场低速徘徊寻找车位时间，日积月累可以减少非必要的私家车碳排放[3]。

1.2 课题国内外研究现状

从国内现状看，大部分停车场管理已经实现自动出入场以及无现金支付，但是能够监控空闲时段，并接受车位预约的停车场却寥寥无几。此外，停车场软硬件解决方案的厂家鱼龙混杂，大量多样的车位资源没有统一的管理平台，难以实现方便用户的共享模式。

共享停车并不是新概念。由一线城市率先颁布，国内部分城市最早在2016年就出台了将政府机关内部停车场错时对外开发的政策，以便提高停车资源共享利用。各大城市对共享停车大力推动，资本涌入共享停车市场，大量停车企业完成融资，其中ETCP共享停车公司目前市场占有最大。然而应为技术问题（计时计费准确度、车位定位和信息准确度）和运营细节问题（异常事故处理、物业车位核验），共享停车解决方按实施起来却是阻力重重。

国外出色案例为欧洲Mobypark共享停车，业务覆盖欧盟中十一个国家地区将近20座城市的酒店、企业和医院等停车场。值得借鉴的是Mobypark为了确保用户和车位拥有者守时守约，采用评论机制，并在用户违规时主动介入处理。

从共享单车的通过国内外案例分析可以看出，共享停车对于车位管理物业和车主是互利共赢的解决方案，可以有效缓解物业角度出发车位资源浪费现象、车主角度出发车位资源稀缺的情况。为了成功开展共享停车解决方案，只有建设更加完美的、符合我国市场需求的停车预约管理平台才能解决现有方案的痛点。

1.3 课题研究内容

通过系统的SWE方法论设计并实现一个共享停车预约管理网站。共享停车平台用户角色有车主、车位拥有者、物业以及管理员。系统采用Vue/JavaScript前端框架以及Flask后端，并且使用MySQL作为数据库软件。后端使用SQLAlchemy操作数据库。地图主要采用了Google Map JavaScript API以及Google Places API。除了前后端开发外，在本设计中考虑到方便实际部署，将前端基于Nginx的Docker镜像进行搭建，后端则将Python后端程序以及数据库分别部署到两个Docker容器中，并开放端口方便调用。系统功能有登录、注册、登出、按车位名查询车位、按目的地的查询车位、查询车位空闲时段、预约车位、增加车位、查看订单列表、入场出场、取消订单、查看车位使用率以及收入统计等。

1.4 论文组织结构

第1部分，介绍实现一个共享停车平台的意义、现状以及内容。

第2部分，本设计中使用到的技术介绍，设计中使用到到工具介绍。

第3部分，共享停车平台的需求分析和最终实现软件的运行环境。

第4部分，讲解系统总体设计，包括结构设计、模块划分、业务上使用流程以及DB构架表的划分；第5五部分内容会对第4部分做详细补充，对UI和交互设计、具体模块功能。

第6部分，根据代码分析，介绍系统功能具体实现方法。

第7部分，从用户交互界面易用性、直观性上进行测试；测试软件整体以及部分的可靠性、鲁棒性。

第8部分，总结本次毕业设计优劣以及从中的学习以及领悟。

# 2 技术及工具介绍

2.1 浏览器获取定位Geolocation API

Geolocation API在用户允许的情况下，用户可以向Web程序提供地理位置。首次使用时用户系统会弹窗获取用户定位权限。使用Geolocation API时调用navigator.geolocation，浏览器会弹窗向用户获许，同意后浏览器将使用设备上最佳可用功能（如GPS）获取定位。原生Geolocation API中基于WI-FI的定位由Google提供，也可以使用百度、腾讯或者Autonavi等第三方提供的基于IP的定位服务[4]。

2.2 Visual Studio Code

本设计集成开发环境（IDE, Integrated Development Environment）使用了非常热门的VS Code。Visual Studio Code是微软开发的源代码编辑软件，支持数百种语言的语句标注、括号匹配、自动缩进。丰富的插件可以快速扩展VS Code的功能。本设计的Web应用程序使用了HTML, CSS, JavaScript, Python, Dockerfile, Shell Command, yml等语言，VS Code都可以胜任他们的编辑、Debug以及运行。此外，内建的Git工具可以方便的进行版本控制（无需使用git指令），内建的Debug工具可以挂载到Docker容器的端口进行监控以及Debug。

2.3 MySQL数据库

MySQL是一个开源的关系(R)数据库（DB）管理系统(MS)。名称中My是联合创始人Micheal女儿的名字，SQL则是Structured Query Language结构化查询语句的首字母缩写。MySQL是Oracle的产品，支持Windows, MacOS, Linux多系统，支持数据量大记录条数多的大型数据库，语句Syntax标准并且支持主流的编程语言。通过B-树或者Hash数据结构建立索引正对不同的搜索条件进行优化，以便达到最佳搜索速度。MySQL由C和C++编写，本设计使用了最新的MySQL 8.0版本。数据库管理上，开发时采用MySQL Workbench方便增删查改以及数据展示，生产环境上则在Docker容器中采用MySQL Shell进行root用户操作、建库等操作。

2.4 Flask

Flask是Python的轻量化Web应用程序的框架，使用Werkzeug路由系统，Jinja2模版引擎这两个依赖库。Flask不包含DB层，因此可以使用任意开发者偏好的数据库软件。

2.5 Docker与容器化

Docker是一个封装、发放以及运行软件程序的平台，使用OS级虚拟提供平台即服务（PaaS）的产品。使用Docker可以将开发出的软件以及运行环境（包含环境变量、依赖包、依赖资源和依赖软件等）一并打包成镜像（Docker Image），以便软件以容器化（containerized）方式在任意Microsoft Windows，macOS，Linux系统的计算机上运行。和虚拟机不同的是，Docker容器更加轻量化，一台计算机上可以轻松部署多个容器化的软件。由于Linux的命名空间支持，从操作系统角度看来软件之间运行相对隔离。由于后端使用了Flask App以及MySQL数据库，是多容器应用，本设计使用了Docker Compose工具。Docker由Go语言编写。

2.6 Nginx

Nginx是一个可以具有反向代理以及负载平衡的web服务器软件，2004年首次发布，使用C语言编写，免费开源。本毕业设计使用了Docker Hub上Nginx 1.15版本基础镜像，再次赞美Docker Hub。Nginx现在已经和Apache并列属于最受欢迎的服务器软件，BOA、Capital One等知名企业都在使用。Docker组建镜像时写入配置文件，指定服务器上页面等资源所在的路径，同时路径也将作为相对访问的root路径。

# 3 需求分析以及系统环境

3.1 系统功能需求分析

进行系统需求分析，就是分析系统需要具备的功能，实现系统之后其性能以及系统的可靠性。在需求分析的步骤，划分出功能模块，相应的API参数规格协定，具体的开发任务。共享停车系统具备以下功能：

1. 用户注册：检查手机号是否已经被注册，注册新用户
2. 用户登录：判定用户是否已经注册，判定密码哈希是否正确
3. 用户登出：清理Cookie和Session Storage中的用户身份校验Token，退出登录，使用户重新登录才能使用平台的各项功能
4. 修改用户角色：修改角色为个体用户或者物业用户。
5. 根据车位名称搜索车位：根据输入的关键词模糊匹配名称中含有关键词的车位。
6. 根据目的地搜索周边车位：在地图上搜索目的地，展示目的地附近的停车场和停车位。
7. 向车主展示车位细节：展示车位空闲时段，车位拥有者，车位名称等信息。
8. 查询车位是否有空：查询用户在车位中所选的时段是否可以预约（进而预约或查询别的时段）。
9. 预定车位：预定用户在车位中所选的时，占用车位被选中的时段，创建相应订单。
10. 订单操作：查看用户的订单列表；取消订单、交易完成后订单支付（未调用支付API，仅做了订单状态改变）。
11. 车位订单管理：展示车位不同状态的订单；拒绝用户订单。
12. 向拥有者展示车位：查看我的车位列表，展示选中日期车位使用率、总订单数
13. 修改车位信息：启用停用车位，修改车位计费费率。
14. 添加新的车位：地图上选定车位所在地址，填写车位相关信息，提交新车位请求
15. TODO admin: 扫码出场入场？身份转换确认或拒绝？

系统性能、可靠性以及易用性：

1. 性能上：设计时采用高效的表设计，可以运行时实时计算的数据不予写入表记录（例如总订单数，总收益），多采用字典这样高效访问的数据结构，在运行时效率和存储读写效率之间权衡；开发过程反复迭代、优化各模块以及接口的算法，必要时对设计进行调整，以便达到能力范围内最优的算法复杂度。
2. 可靠性上：在前端与后端进行了大量非法输入的检测以及报错处理，杜绝了由于用户提交的内容不规范造成的程序异常。例如在前端进行手机号码格式的验证、数字的正则表达式验证等。后端则提供了丰富的错误码，考虑方方面面用户操作不当的情况，从而从用户使用角度杜绝程序异常。对错误码进行再次封装，均返回200 Success由前端处理，这样可以防止500 Server Error报错码引发的服务器端源代码泄露。测试数据精心设计，考虑不同使用情况，不符合常理的使用也考虑在内。多轮测试和加强后程序可靠性和鲁棒性都非常出色。
3. 易用性上：本毕业设计将相当一部分大的时间投入在前端页面设计以及美化上。一个有优秀功能的网络平台，如果没有用户友好的操作界面，也将在吸引用户上失败。因此共享停车平台UI设计精美简洁，直观易用。用户可以很直观地找到并使用想要的功能。同时CSS美化部分加入大量缓入缓出动画、页面元素阴影、元素圆角、模糊和透明度等设计元素，提升了页面的质感与高级感。

3.2 可行性分析

1. 技术可行性

技术栈采用了主流的Vue+Flask+MySQL on Docker的前端、后端、数据库和部署项目构架。前端部署在容器化的Nginx服务器软件，后端则采用了一个多容器（Flask容器+MySQL数据库容器）的Docker应用程序。地图采用了Google Map API和Google Places API服务，才Google Cloud Platform上创建项目后就可以启用者两个API。数据库访问使用了Python SQLAlchemy包。开发工具使用Visual Studio Code，免费且有丰富的插件扩展其对各种语言的支持和功能扩展，提高了开发效率和开发体验。在依赖上，前端引用了md5.js生成密码Hash，axios.js HTTP client向后端服务器发起请求，jQuery覆盖没有使用Vue的情况让HTML元素选取更方便；后端依赖SQLAlchemy, PyJWT等包。使用的技术热度较高，因此比较方便寻找文档以及碰到问题时可以在互联网找到解决方案。

（2）经济可行性

我国机动车现状是保有量大且增速快，停车位缺口大，因此共享停车平台市场是巨大的。开发使用了许多免费、开源的工具以及软件，减少了开发成本。Google Cloud Platform的API计费根据调用次数计算，因此不会造成生产环境使用外额外的收费。最后将软件平台均做了容器化处理，因此不需要部署在专门的服务器上，Docker实现软件间的隔离，减少了运营成本。其中调用二维码的API由goqr.me免费提供。共享停车由于提高了车位利用率，可以有效减少每车位每人次停车费率。从车主角度看来停车不仅更加方便而且更加省钱，从车位拥有者角度看来，闲置车位可以出租产生收益、提高总收益，实现共享经济互利共赢。

（3）运行可行性

共享停车Web平台采用了C/S（Client/Server）的软件结构，因此客户使用浏览器就可以访问共享停车平台，由于CSS编写时正对Chrome浏览器，因此建议通过Chrome访问共享停车平台。前后端均采用Docker容器化处理，可以在任何安装有Docker软件的宿主机上方便地部署与运行。共享停车平台毕业设计中使用到的API均由互联网企业提供，运行时调用服务可靠稳定。

3.3 系统环境

开发系统：macOS, Linux

开发IDE以及工具：Visual Studio Code, Hbuilder, Postman, MySQL Workbench

软件语言：Python, JavaScript, Shell, TXT以及YAML配置文件

Python版本：开发环境 3.8.3, 生产环境3.8-slim

MySQL版本：开发环境8.0.26，生产环境8.0.26

Nginx版本：Docker镜像1.18.5-alpine

生产环境：任意装有Docker的宿主计算机，部署在对公网开放端口的Linux Server上最为理想

# 4 系统总体设计

通过需求分析得出系统结构设计，业务逻辑以及数据库设计（主要包括概念设计和逻辑设计）。本部分介绍整体看来的系统构架，并从业务流程上介绍共享停车平台的设计，最后将平台的数据库结构设计和逻辑设计。

4.1 系统功能结构设计

共享停车业务的需求分析中，得出共享汽车平台系统总构架。Web应用平台将包括个人用户、物业用户以及管理员三种角色。通用的功能有注册、登录、退出登录和角色切换。个人用户的功能将有多种方式寻找车位、查询车位、预定车位、管理预定（订单）、以及个人所有车位的管理；物业用户的功能紧密围绕车位管理，具体有车位发布、车位信息概览、订单概览、具体订单操作；管理员功能可以管理用户。总体上开来系统功能图示如下图4-1。



图4-1 系统总体结构图

平台具体功能如下：

1. 注册：检查用户是否已经存在，注册新账户，默认角色为个人用户。
2. 登录：检查用户是否存在，检查密码哈希值是否匹配，生成加密Token返回给用户浏览器，以便后续请求使用。页面根据不同的用户角色跳转到相对的用户主页面。
3. 退出登录：清理浏览器SessionStorage以及Cookies中存放的信息，清除Token返回登录页面。
4. 改变角色：在用户，物业之间切换角色，会引起重新登录。
5. 寻找车位：共享停车平台两种方式寻找车位。通过车位名称搜索，输入关键字匹配车位名；选择目的地，寻找目的地周边范围内可用车位。
6. 查询车位可用性：查询指定车位在指定时间是否空闲。
7. 预定车位：预定指定车位的某个时间段。
8. 订单管理：在预约时间开始前可以的操作有取消订单，预约时间开始后可以入场、离场，离场后可以支付订单。
9. 添加车位：个人和物业角色的用户均可以添加自己的车位。
10. 车位订单管理：可以获取每一天车位订单收益以及使用率概览，也可以获取总览；可以管理车位订单，拒绝用户使用车位的预约。
11. 车位管理：包括启用停用车位，修改车位费率。

以下功能仍在打磨，有限时间内无法将业务逻辑做完美：

1. 停车场管理：针对物业用户批量管理停车位的需求，拟实现物业用户特有功能停车场管理，对停车场的所有单独车位资源进行抽象，对外看到物业停车场新式操作管理，开发仍在进行中。
2. 管理员功能：目前管理员管理管理用户、管理车位与停车场、管理订单使用方式是原始的且没有GUI操作的方法。作为开发者，可以在生产服务器上ssh到MySQL容器中对表进行操作。或使用MySQL Workbench连接至Docker容器映射到宿主机的端口上，用更加用户友好的方式进行管理。
3. 扫码入场出场：订单生成唯一标识，将标识嵌入二维码，扫码即可实现入场以及后续出场功能，目前 拟把功能赋予物业用户。

4.2 共享停车平台功能以及业务流程设计

4.2.1 登录注册流程

本毕业设计的共享停车平台几乎所有的功能都需要用户登录后使用，用户没登录前可以访问网站主页查看宣传页面。因此后续功能默认考虑用户已经登录的情况，不在对登录进行赘述。新用户注册，注册后跳转至登录界面，重新输入密码。登录成功后根据用户角色跳转到不同对用户主界面。不同角色用户主界面包含不同功能。

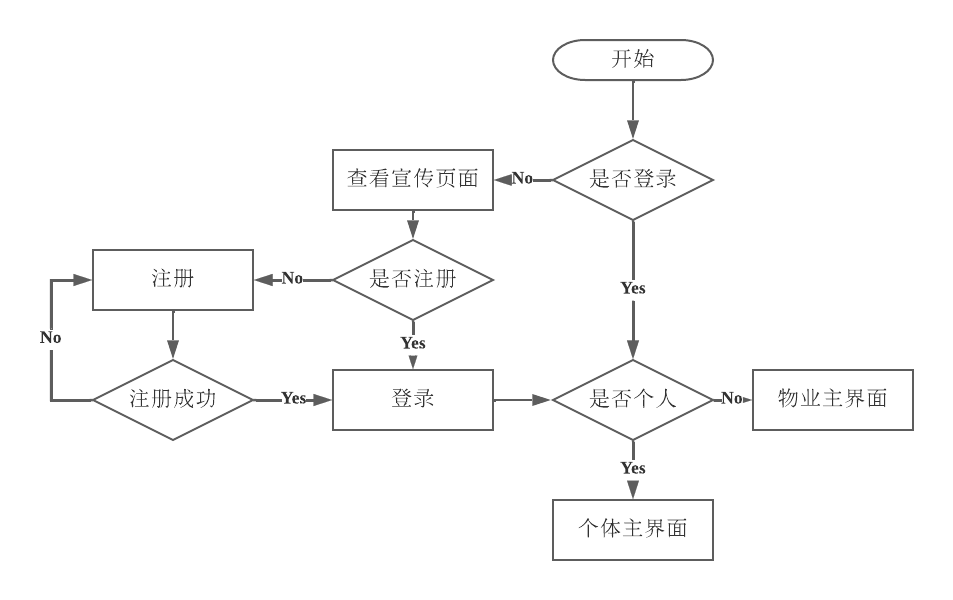


图4-2 共享停车平台登录注册流程图

4.2.2 用户主界面操作流程

用户登录后，进入平台主页面。个人用户主页面默认子页面为寻找车位，物业用户主页面默认为车位管理。在主页面可以选择针对不同业务功能的子页面。具体操作逻辑以及使用流程如下图所示。

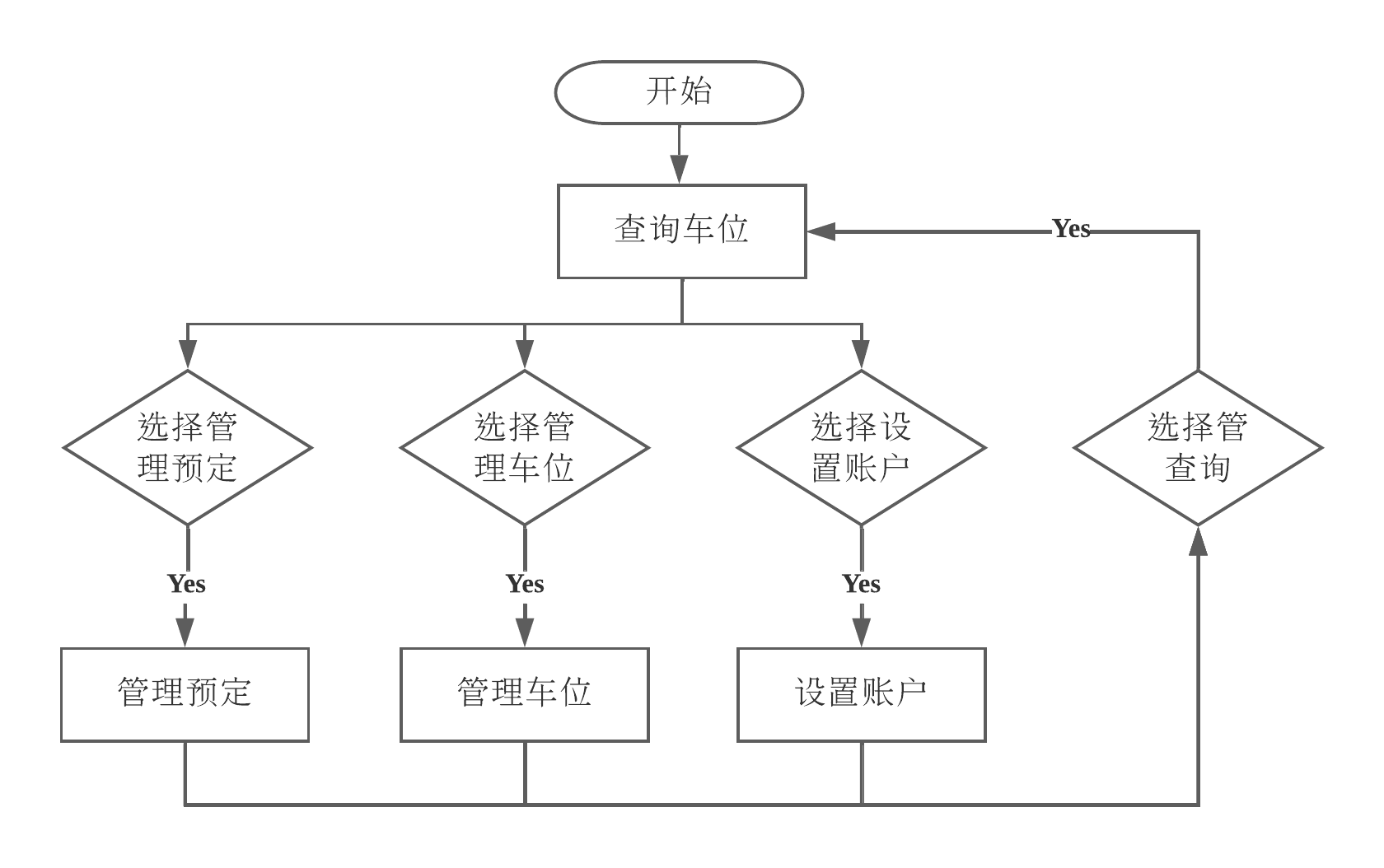


图4-3a个人角色主页流程图

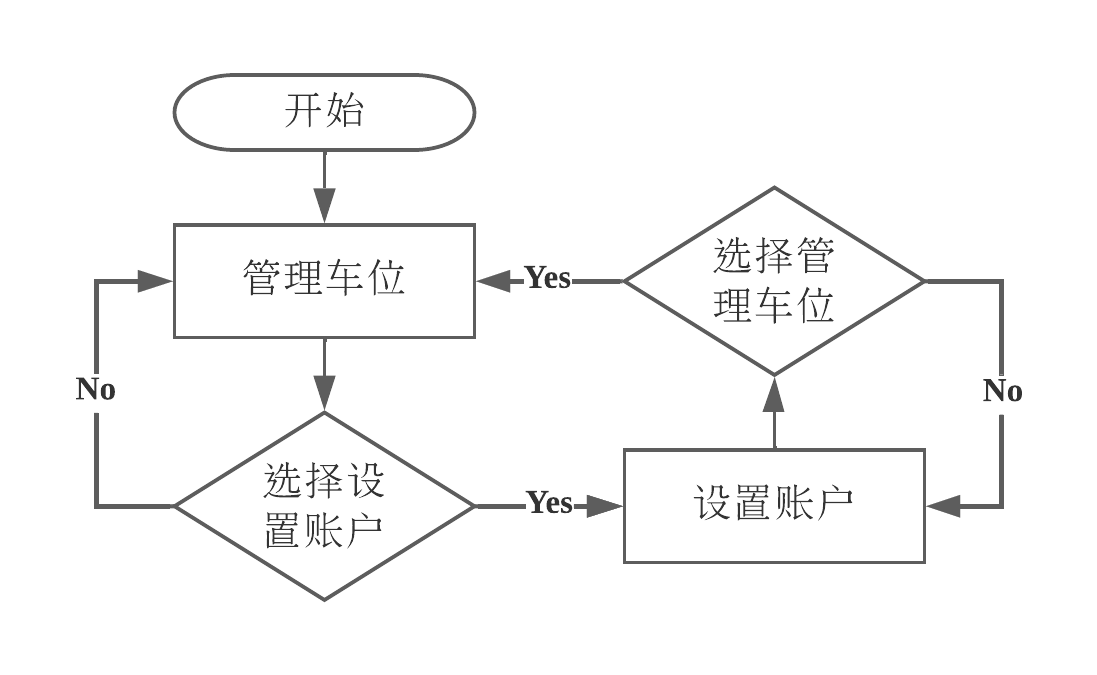


图4-3b物业角色主页流程图

4.2.3 关键字车位搜索流程

用户在搜索车位页面有两种方式搜索车位，根据车位名搜索以及在目的地附近搜索。本小节介绍输入关键词匹配车位名搜索车位的方式。用户输入关键字（例如“苏州中心”），按下搜索按钮，返回车位名称中含有关键词的车位，并根据离用户的距离排序。

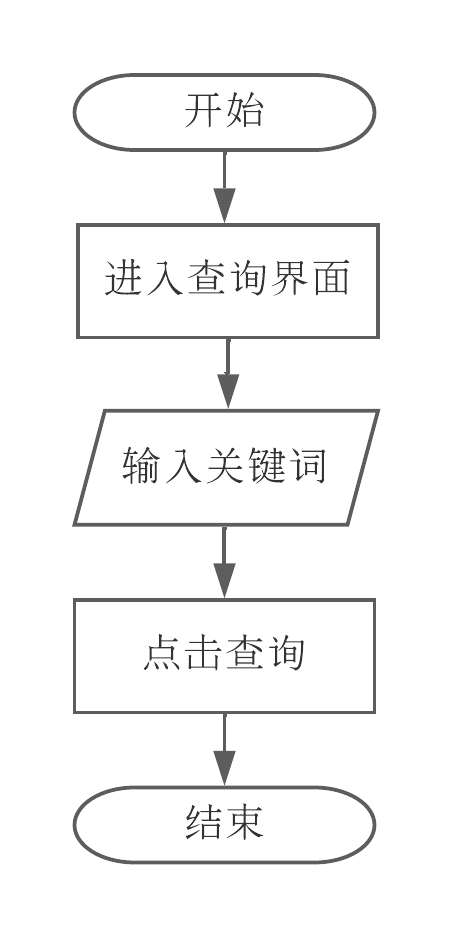


图4-4按照车位名称关键词搜索车位流程图

4.2.4 查询目的地周围的车位搜索流程

用户另一种查找车位资源的方式是搜索目的周边的车位。用户在地图上的输入框输入目的地名称，在弹出的目的地列表中选取具体目的地，系统会查询并返回选中目的地周边范围内（默认2公里范围内，后续完善会加入筛选选项）的车位。

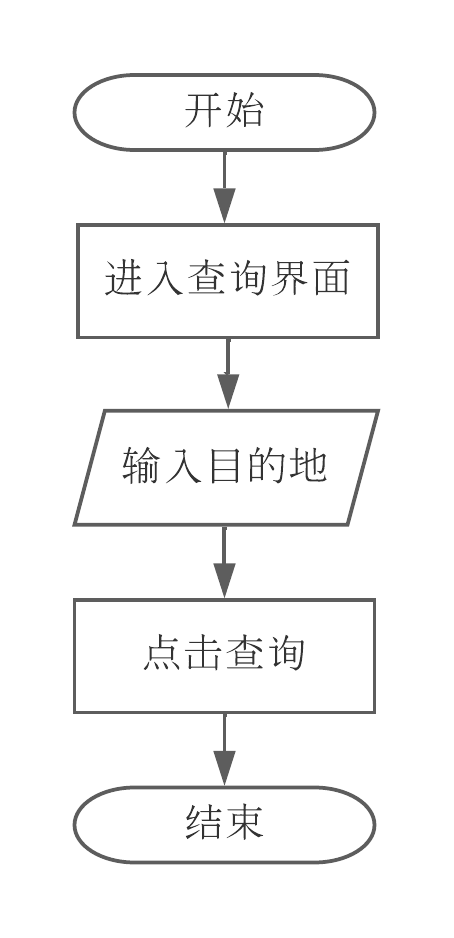


图4-5 查看课程流程图

4.2.5 查看车位-查询可用性-车位预约详情流程

用户在共享停车平台上以以上两种查询车位后，点击返回列表或者地图上的车位，将展示车位详情。直观的图形界面将以直观的方式展示给用户：车位在选中的日期中可用的时间段、车位属于个人还是物业、车位收费费率以及车位名。选择结束开始日期与时间后，查询时间段车位可用性，即可预约。

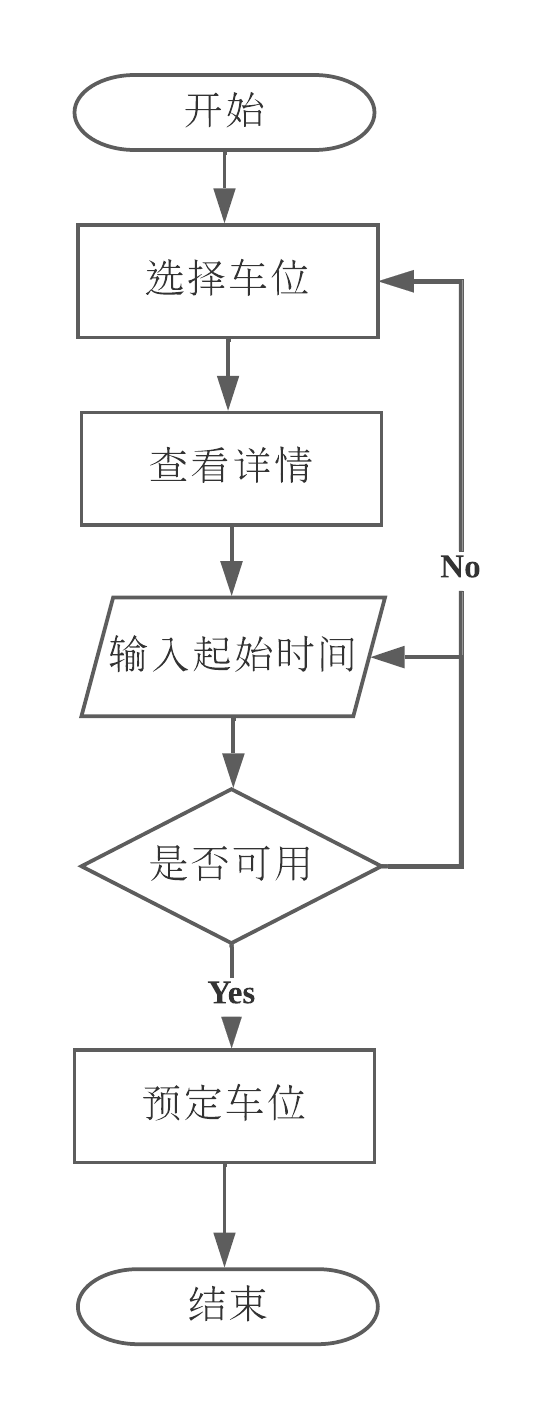


图4-6 查看详情并预定于流程

4.2.6 订单管理逻辑与流程

个人用户从主页面目录中选择订单管理，进入订单管理二级页面。系统将给用户展示用户的订单，包括订单开始结束时间，订单状态，订单价格等信息。订单状态有尚未开始，使用中，待支付，已完成。在预约时间开始前可以取消订单。到达预约时间后可以入场，在结束时间前出场，离场后可以支付订单。未使用且未取消的订单会列入异常订单。

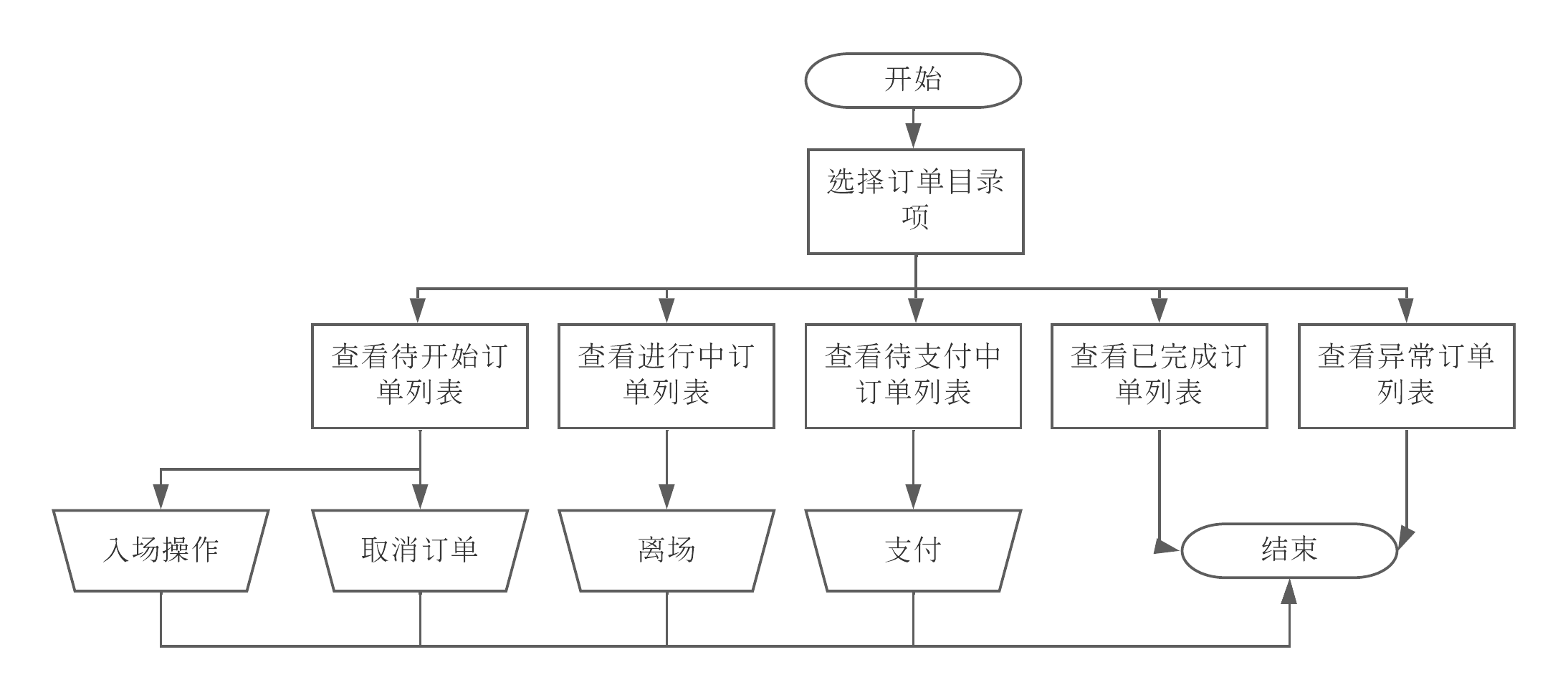


图4-7 个人用户订单管理流程

4.2.7 （通用）用户设置账号流程

用户主界面目录中选择账号设置，进入账号设置二级菜单。账户设置页面中可以看到用户姓名以及手机号。个人用户可以选择申请成为物业用户。用户在此界面可以退出登录。

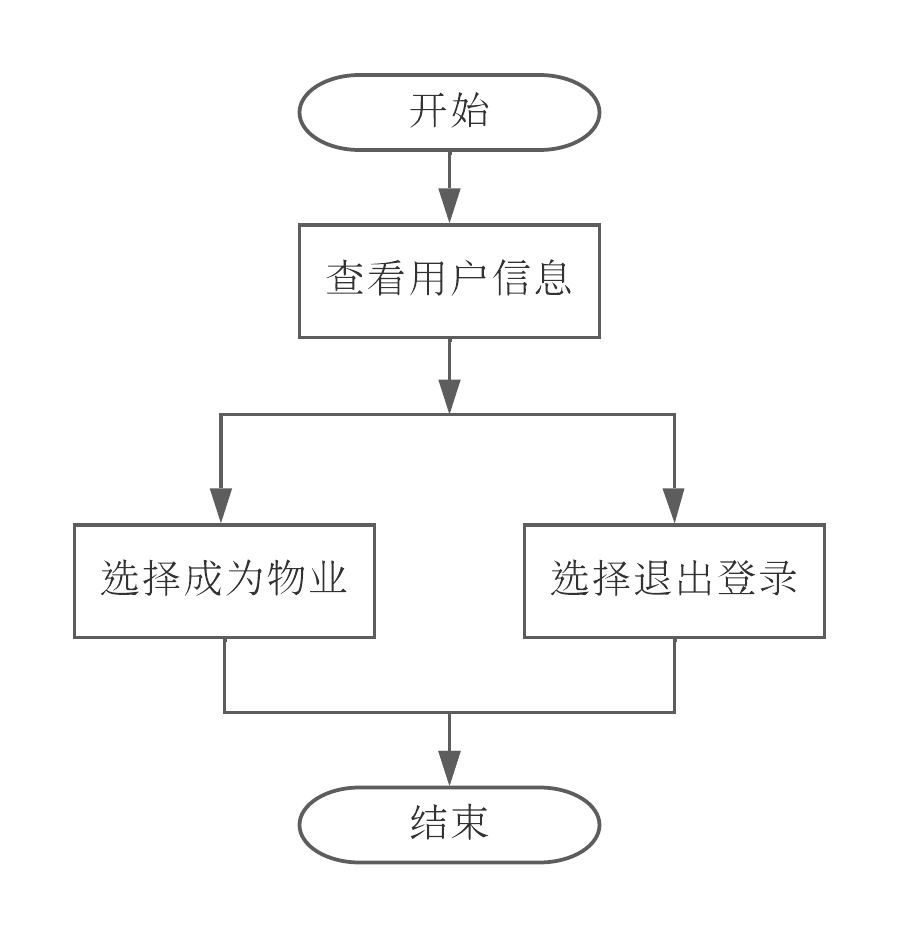


图4-8 账号管理流程图

4.2.8 （通用）车位管理之获取车位概览流程

业主在小区购置车位的情况普遍存在，因此允许他们也将车位资源共享出来。用户在主界面侧栏菜单选择进入车位管理二级页面。车位管理页面展示用户发布的所有车位，并做有车位总数、订单总数、活跃订单数统计数据。也可以选择日期，查看指定车位当天的统计数据。

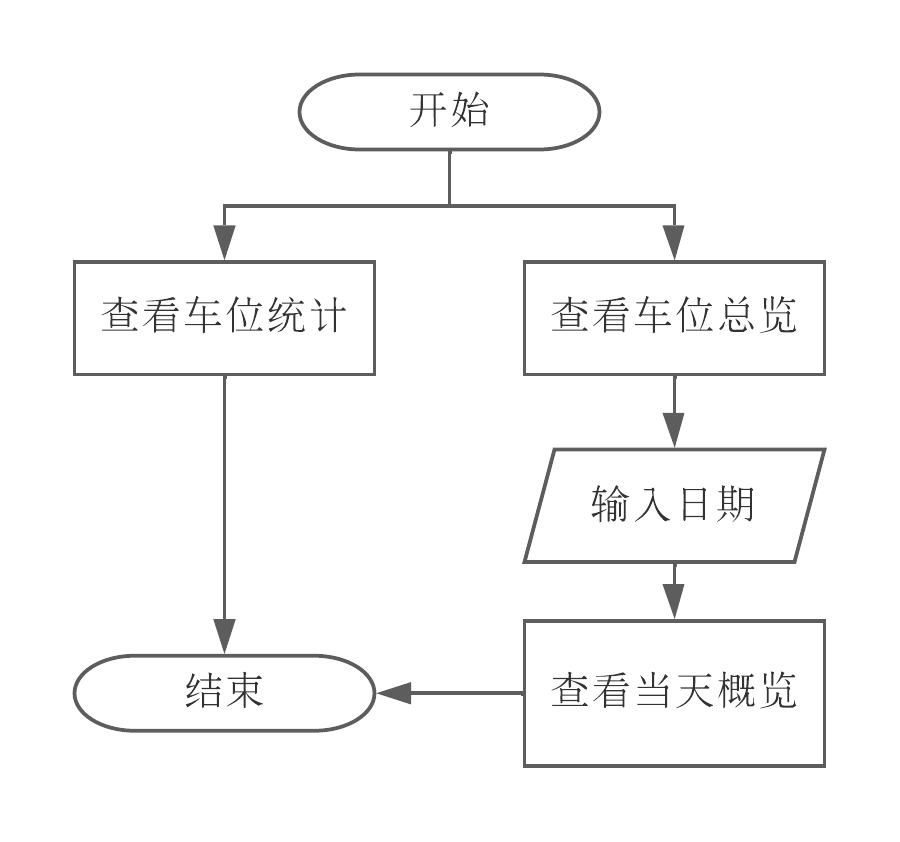


图4-9 车位概览流程图

4.2.9 车位订单管理流程

用在上一部分车位概览流程中，选择查看车位详细信息即可管理车位所属订单。订单管理可以查看不同状态的订单（已完成，未开始，进行中，待付款以及其他状态）并且对他们做出相应的管理操作。

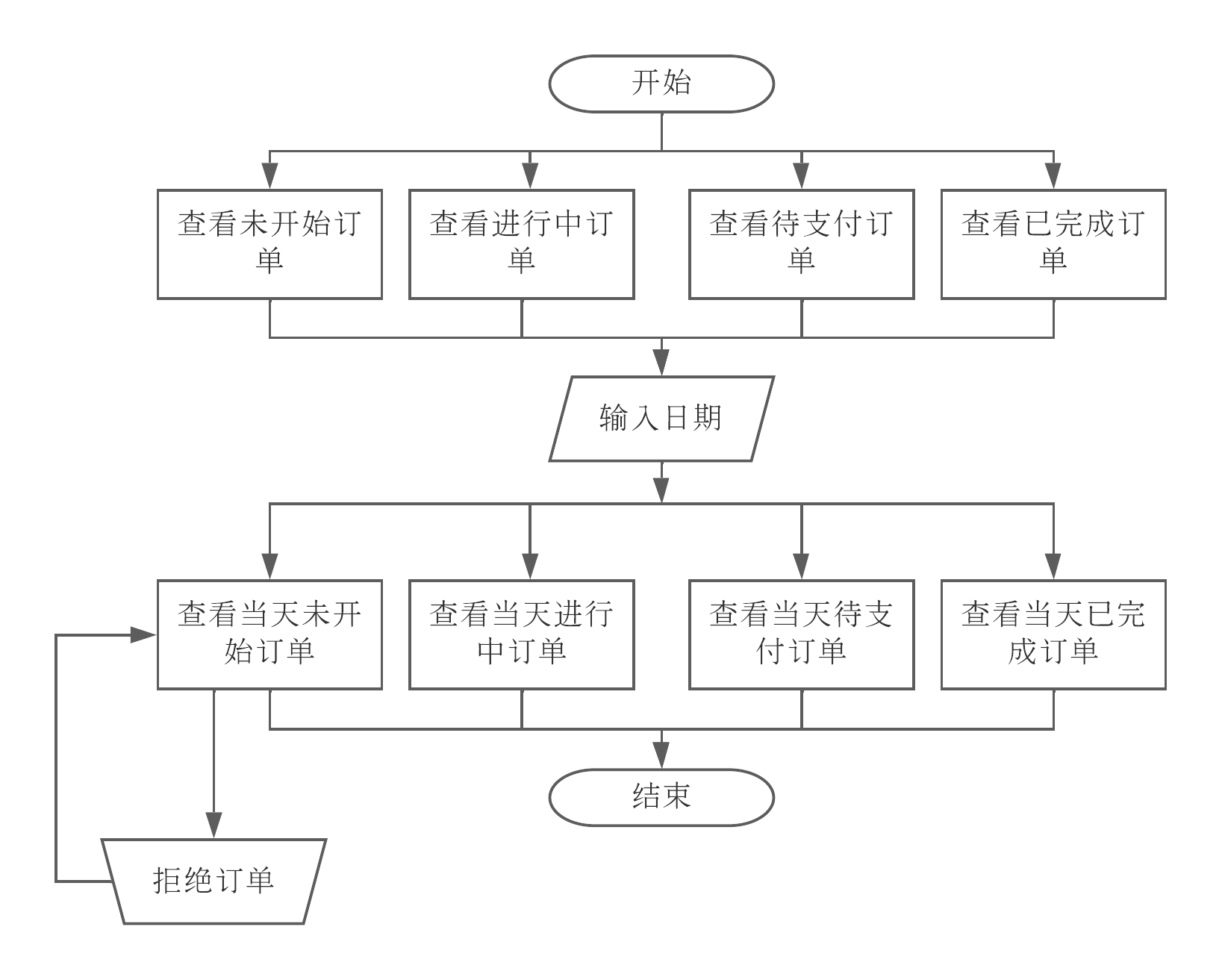


图4-10 车位订单管理流程图

4.2.10 添加车位流程

审核通过的机构账户可以用列表的方式查看已有的课程信息，也可以创建课程并上传视频，已经创建好的课程和上传的视频信息可以被机构用户修改。课程管理流程图如图4-11所示。

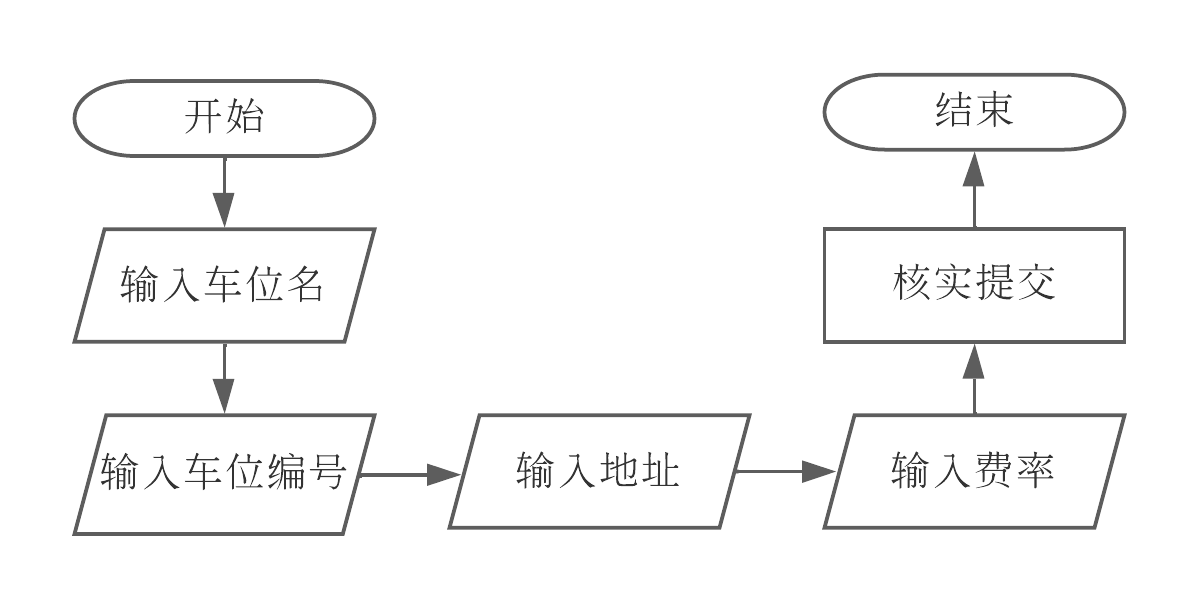


图4-11 添加车位流程

4.2.11 修改车位属性流程

车位拥有车（包括个体用户以及物业用户）在车位详细信息页面可以对车位费率进行修改。用户也可以开发使用车位或停用车位并不再接受新订单。

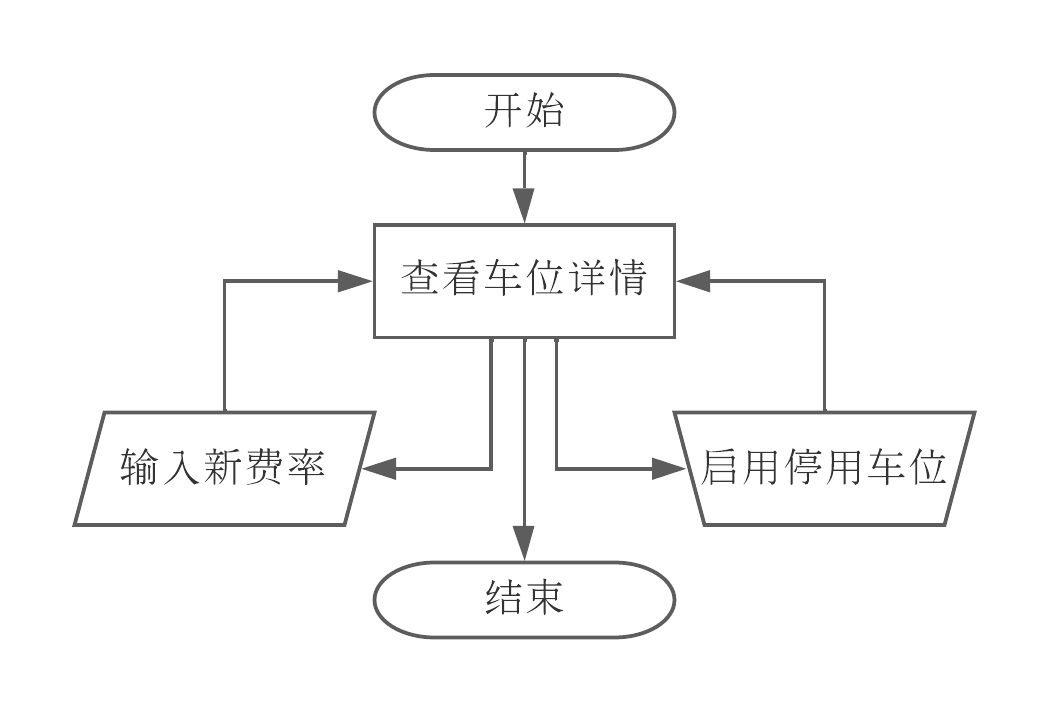


图4-12 车位属性设置流程

4.3 数据库设计

数据库是任何多用户Web应用程序的核心。从本毕设开发者的浅薄开发经验体会看来，做好数据库的概念设计以及数据库的逻辑设计，不仅是实现软件预期业务功能的必要条件，设计的合理性以及优化程度更是决定了基于数据的处理逻辑开发难度、软件运行效率和管理维护效率。本小节基于软件业务（幻想我既是项目经理也是程序员，从项目经理角度带入思考，希望借此提高项目的质量）实际需求，在存储开销以及计算开销之间权衡所进行的数据库逻辑以及概念设计。

4.3.1 概念设计

通过需求分析，从现实世界抽离出需要管理的概念到信息世界，得出业务基本逻辑需要管理的数据实体如下：用户、停车位、停车场以及订单。以下将讲述不同实体以及其的属性，并在需要的地方讲述设计思路。

（1）账号信息实体

用户实体涵盖了现实生活中的车主，物业以及管理员。他们都由系统用户这个基本实体派生出，所以归为一种实体管理。各角色之间由角色类别这一属性作区分，其他属性共享。这样以运行时判断的计算开销换取存储开销的节省，而且方便了角色切换的操作。属性包括手机号，姓名，密码哈希，用户类别，创建时间以及信用分。用户实体E-R图如4-13所示。

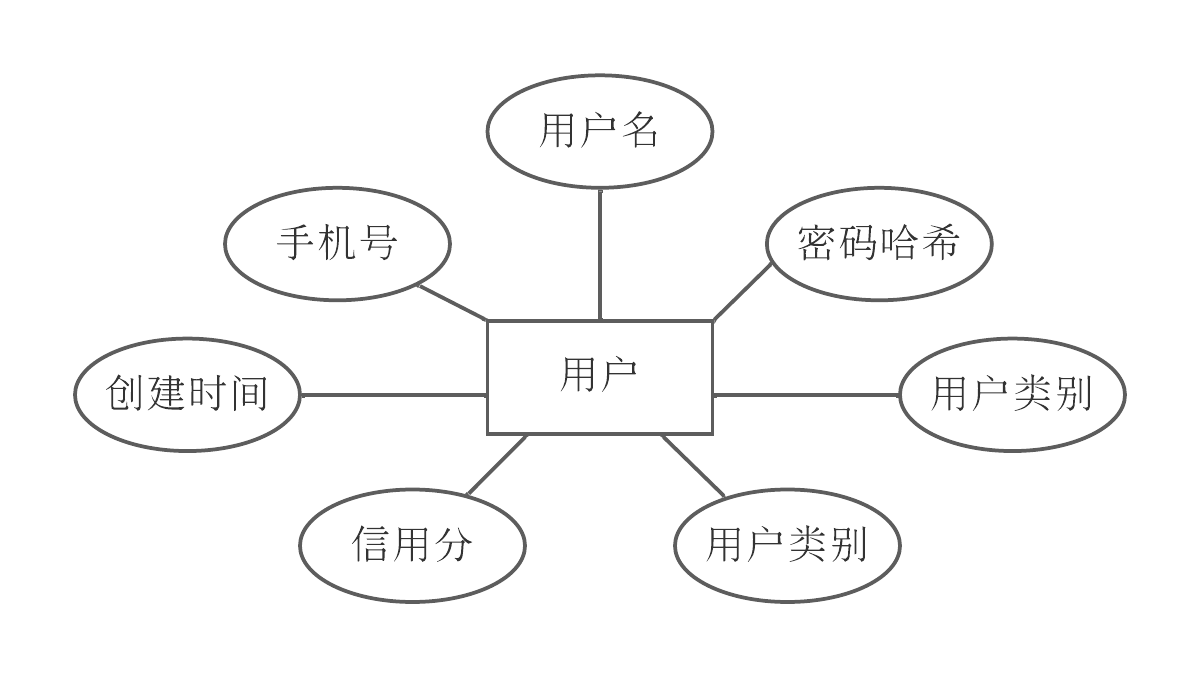


图4-13 用户实体属性图

（2）车位实体

现实世界中的车位抽象成为车位实体。其属性包括ID，车位名，车位类型（属于停车场还是独立车位），拥有者电话号码，车位费率，可用状态，坐标，以及已经被预约的时间。搜索车位的功能将围绕车位实体展开。抽象出的停车位实体E-R图如4-14所示。

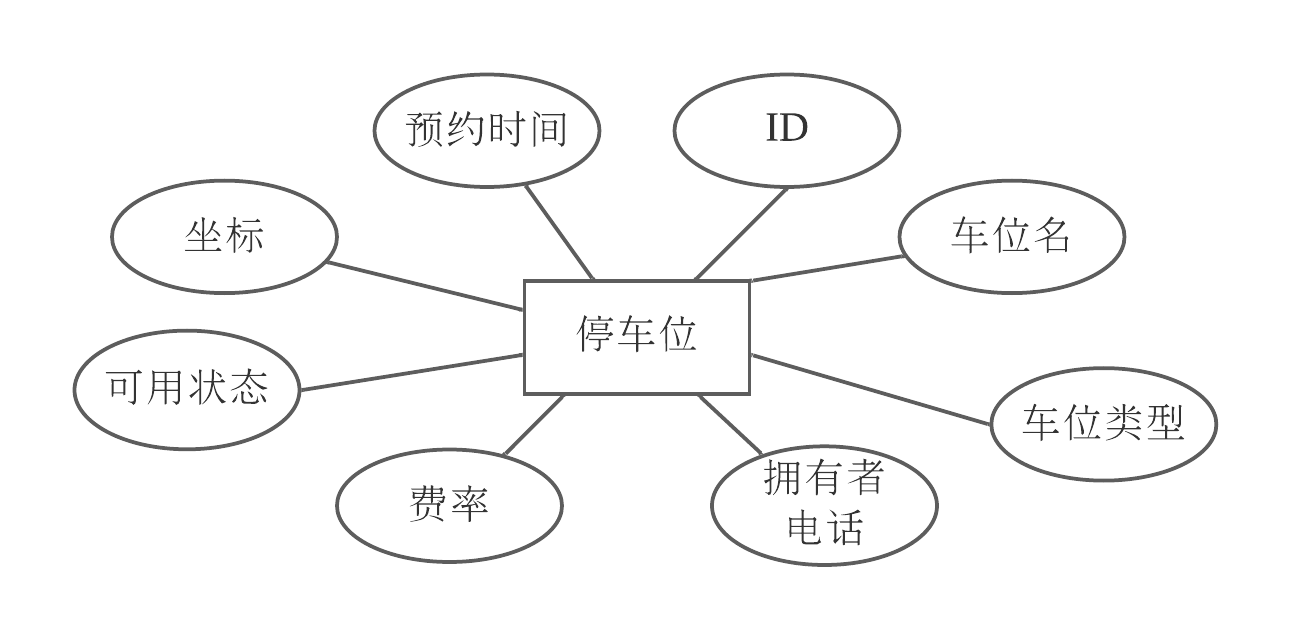


图4-14 停车位实体属性图

1. 停车场实体

现实世界中的停车场抽象成为停车场，停车场可以包含众多停车位。停车位将使用车位类别以及所属停车位ID标识从属关系。停车场实体包含的属性有ID，停车场名称，物业电话，费率，坐标，可用状态。停车场实体属性图见图4-15。

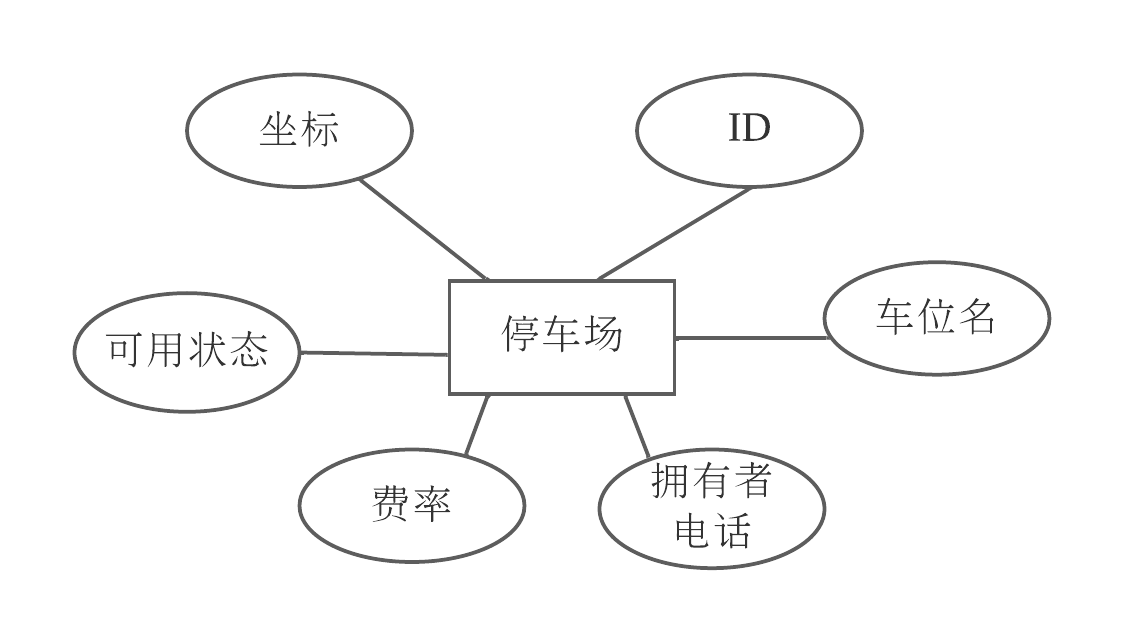


图4-15 停车场实体属性图

（4）订单实体

用户预定停车位产生订单，将其抽象为订单实体。订单属性包括：订单号，客户手机号，车位号，订单状态，创建时间，完成时间，预约开始时间，预约结束时间，实际开始时间，实际结束时间以及下单时的车位费率。订单E-R图如下所示。

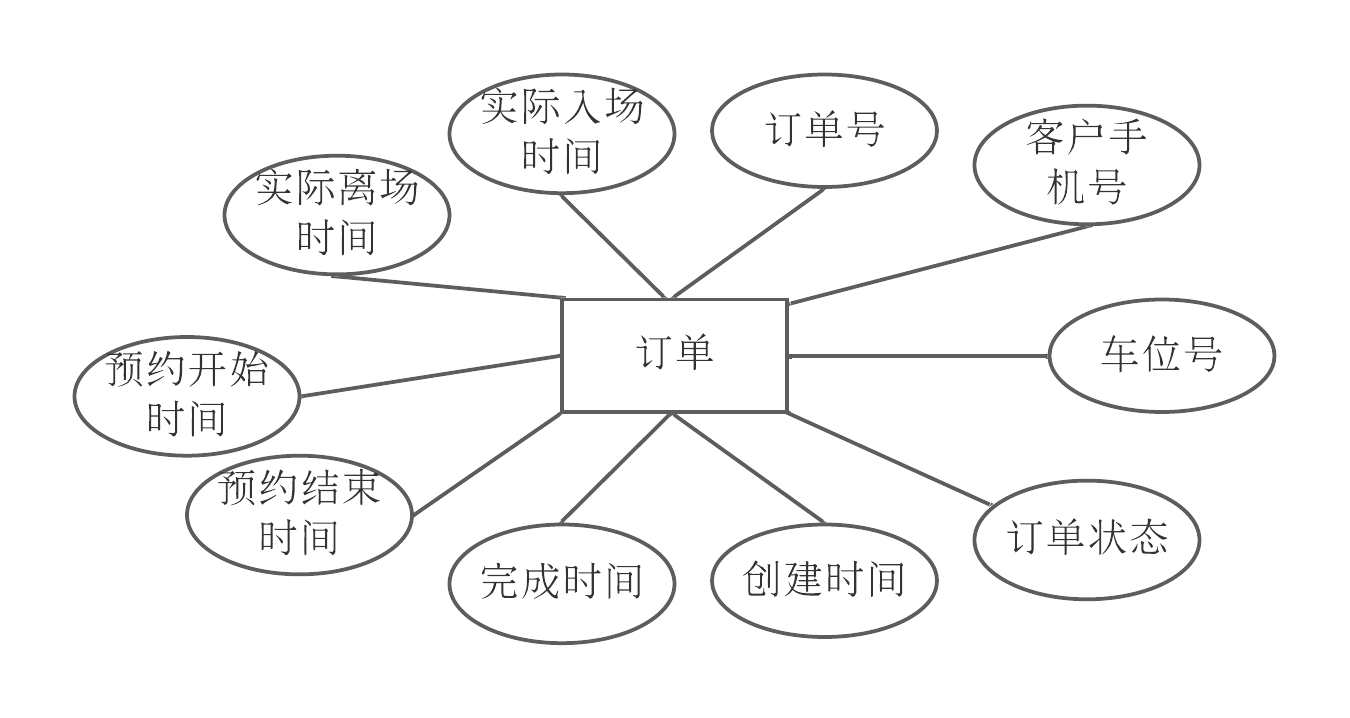


图4-16 订单属性图

1. 实体总览

这一部分分析实体之间的联系。通过分析现实世界业务结合实际概念设计抽象出的实体，可以得出实体之间的联系：

1. 一个个人用户可以拥有多个订单，一个订单只能有一个客户，因此用户实体与订单实体之间存在1对多的关系；
2. 一个停车场可以包括许多停车位，然而一个停车位只能隶属一个停车场，因此停车场与停车位之间为1对多的关系；
3. 一个用户可以拥有多个停车场/停车位，一个停车场/停车位只能有一个拥有者，因此用户实体与停车场/停车位之间存在1对多的关系；
4. 一个停车为可以产生多个订单，然而一个订单只能包含一个停车位，因此订单与停停车位之间为1对多的关系；

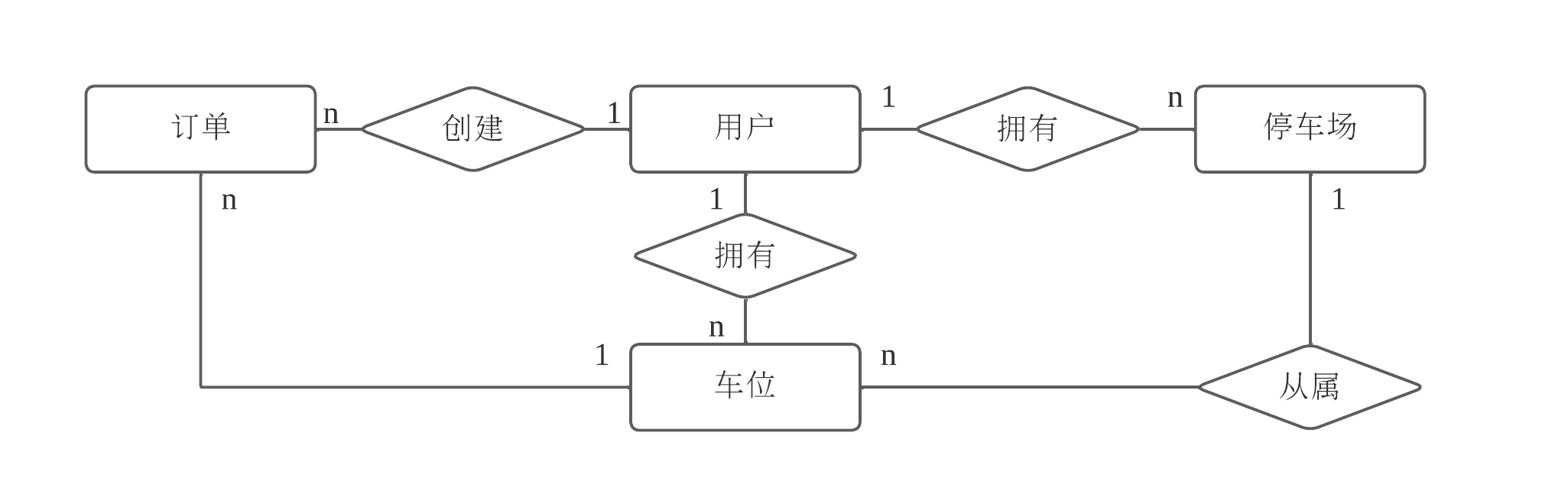


图4-17 系统E-R图

### 4.3.2 逻辑设计（Table/Schema）

（1）user

用户表包括用户手机号（由于手机号和身份绑定，因此将其作为用户唯一标识，是用户表的主键PK），用户实际姓名，32位字符串密码哈希，用户类别（用户类别常数后端定义1个人用户，2物业用户，3管理员）以及创建日期。

表4-1 用户表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column | Type | Length | NN | PK | FK | 描述 |
| tel | varchar | 11 | Y | Y |  | 用户编号，UUID4 |
| name | varchar | 12 | Y |  |  | 用户合法姓名 |
| password\_hash | varchar | 32 | Y |  |  | 密码md5哈希值 |
| user\_type | int | N/A | Y |  |  | 用户类别，整数常数 |
| create\_date | datetime | N/A | Y |  |  | 注册日期 |
| credits | float | N/A | Y |  |  | 信用度 |

（2）parking\_spot

车位表描述某个具体可以停放一辆车的车位。车位表包括信息列有车位编号，车位名称，车位类型（1属于个人，2属于停车场），车位拥有者手机号，所属停车场编号，每分钟费率，状态，纬度，经度，已经被预约的时间段。

表4-2 车位表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column | Type | Length | NN | PK | FK | 描述 |
| ps\_id | varchar | 36 | Y | Y |  | 停车位编号，UUID4 |
| name | varchar | 255 | Y |  |  | 车位名 |
| spot\_type | int | N/A | Y |  |  | 车位类别常数 |
| owner\_tel | varchar | 11 |  |  | Y | 拥有者手机号 |
| pl\_id | varchar | 36 |  |  | Y | 所属停车场编号 |
| price\_per\_minute | float | N/A | Y |  |  | 每分钟费率 |
| status | int | N/A | Y |  |  | 是否可预约状态 |
| latitude | float | N/A | Y |  |  | 纬度 |
| longitude | float | N/A | Y |  |  | 经度 |
| appointments | json | N/A | Y |  |  | 已经预约的时间段 |
| flag | int | N/A | Y |  |  | 互斥访问标记位 |

（3）parkings\_lot

停车场表记录停车场的各项信息，包括停车场编号，停车场名称，管理者手机号，费率，状态，纬度与经度。

表4-3 停车场表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column | Type | Length | NN | PK | FK | 描述 |
| pl\_id | varchar | 36 | Y | Y |  | 停车场编号，UUID4 |
| name | varchar | 255 | Y |  |  | 车位名 |
| manager\_tel | varchar | 11 | Y |  | Y | 管理者手机号 |
| price\_per\_minute | float | N/A | Y |  |  | 每分钟费率 |
| status | int | N/A | Y |  |  | 是否可预约状态 |
| latitude | float | N/A | Y |  |  | 纬度 |
| longitude | float | N/A | Y |  |  | 经度 |

1. Order

订单表包括订单的各种关键信息，表项有订单编号等如下表所示。

表4-4 订单表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column | Type | Length | NN | PK | FK | 描述 |
| order\_id | varchar | 36 | Y | Y |  | 订单编号，UUID4 |
| custom\_tel | varchar | 11 | Y |  | Y | 客户手机号 |
| ps\_id | varchar | 36 | Y |  | Y | 车位编号 |
| order\_status | int | N/A | Y |  |  | 订单状态 |
| utc\_create\_time | datetime | N/A | Y |  |  | 订单创建时间 |
| utc\_complete\_time | datetime | N/A |  |  |  | 订单结束时间 |
| assigned\_start\_time | datetime | N/A | Y |  |  | 预约入场时间 |
| assigned\_end\_time | datetime | N/A | Y |  |  | 预约离场时间 |
| actual\_start\_time | datetime | N/A |  |  |  | 实际入场时间 |
| actual\_end\_time | datetime | date |  |  |  | 实际离场时间 |
| flag | int | N/A | Y |  |  | 互斥访问标记 |
| price\_per\_minute | float | N/A | Y |  |  | 下单时每分钟费率 |

# 5 系统详细设计

5.1 系统架构设计

5.1.1共享停车平台构架-前端

共享停车平台构架使用HTML，Vue，Axios，JavaScript以及CSS美化。顶层页面使用了Vue实例（生命周期如图5-1），大量动态加载的内容使用JavaScript而非Vue Component以便提高页面响应速度。页面元素设计美化参考codepen.io上优秀前端设计师的设计。前端页面完成后部署至容器化的Nginx镜像（构架图5-2）中。前端发送请求Requet与处理响应Response由Axios HTTP Client实现。

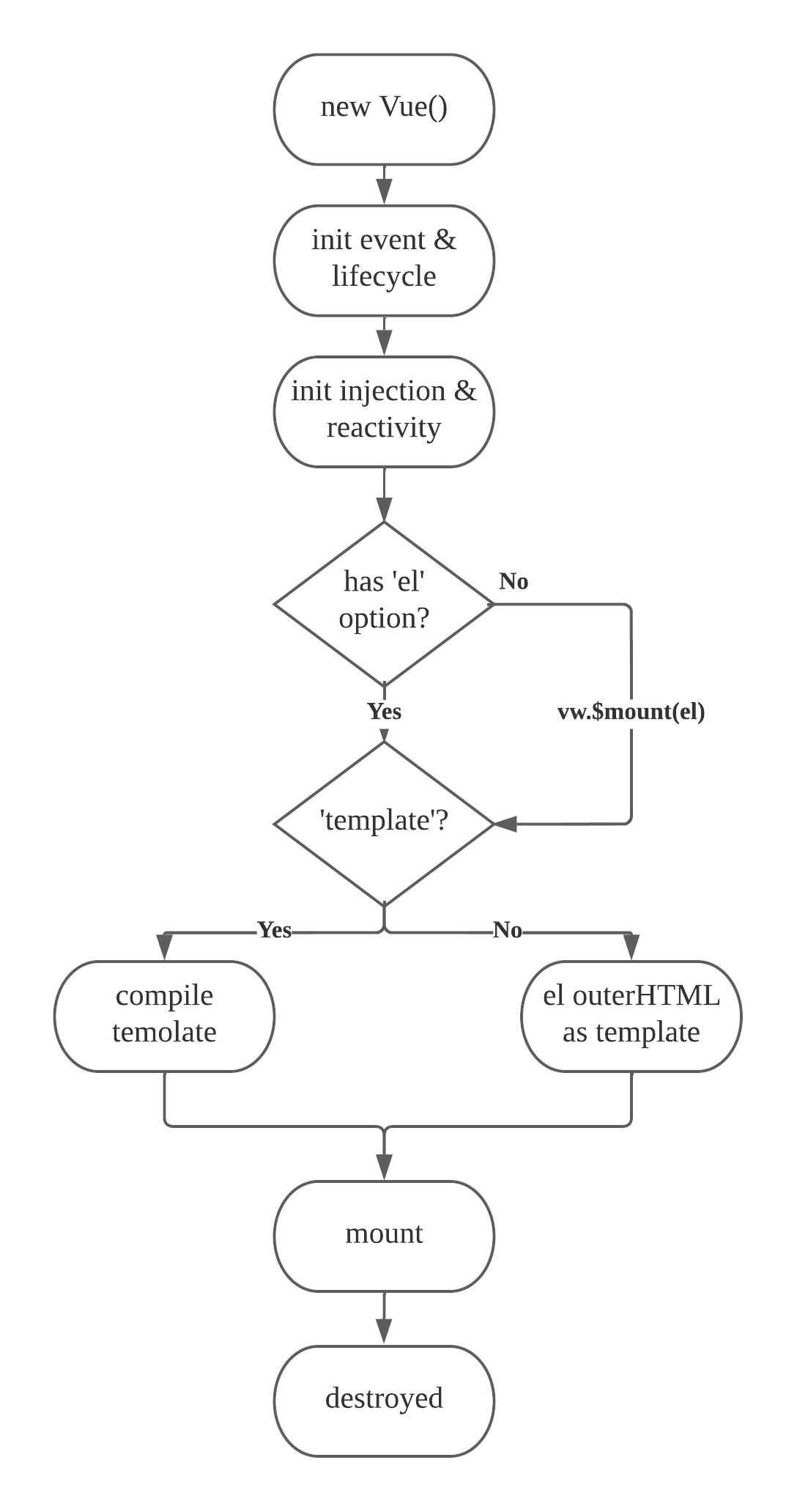


图5-1 Vue实例生命周期

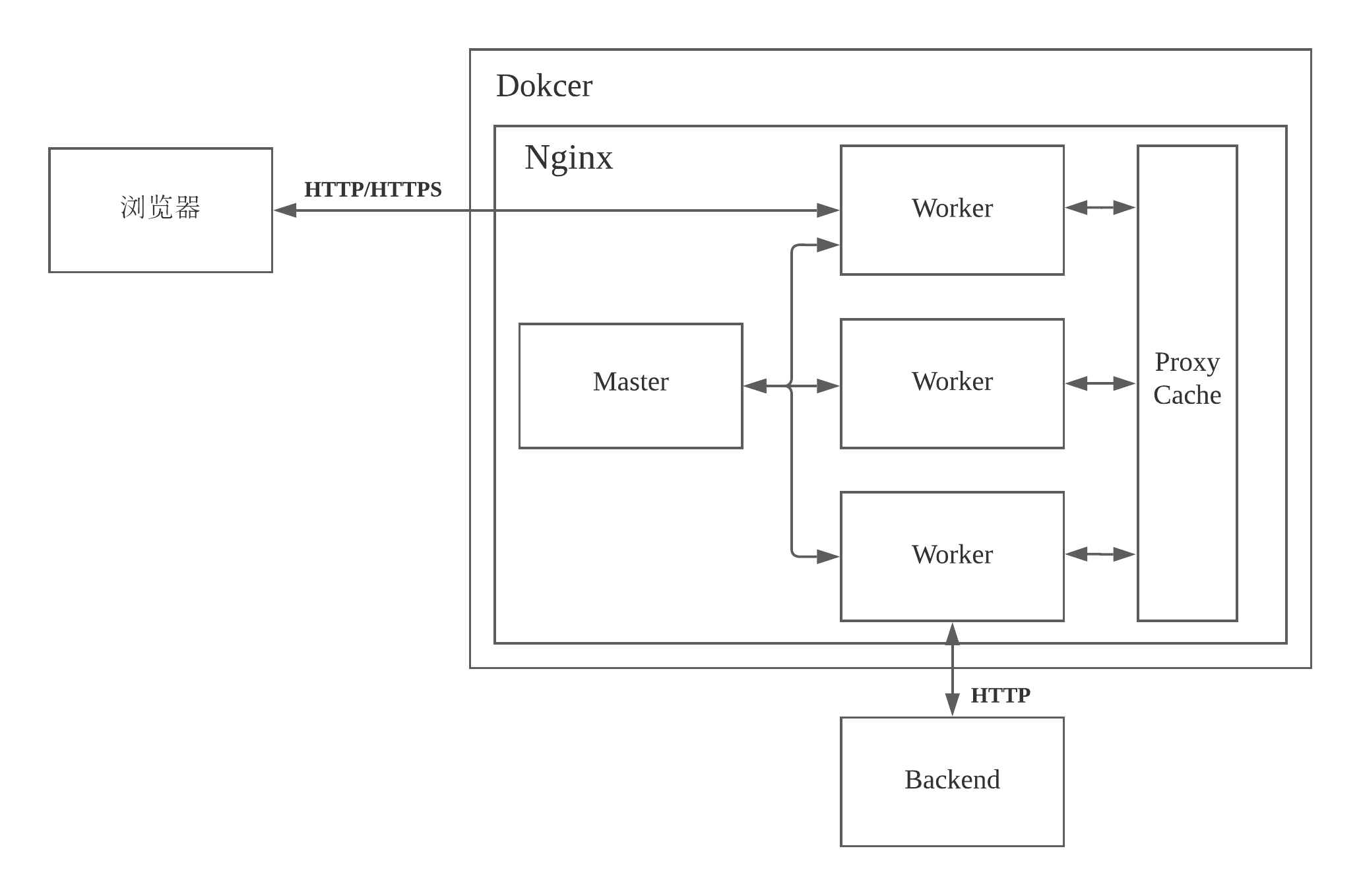


图5-2 容器化Nginx构架（主从负载平衡配置下）

5.1.2 共享停车平台构架-后端

共享停车平台后端使用Flask + Flask SQLAlchemy框架。Flask依赖Werkzeug作为Web服务器网关接口（WSGI，Web Server Gate Interface），作为Python程序与服务器之间的接口，监听服务器指定端口接受TCP传来的HTTP请求。有了网关接口，开发后端时可以忽视网络层包的处理，只在应用层写API。关系对象映射（ORM，Object Relational Mapping）则使用了SQLAlchemy链接MySQL数据库服务进行增删查改。SQLAlchemy既提供了接口，也可以自己编写并执行SQL语句。在共享停车平台最终的部署上，选择阿里云的弹性计算Linux系统服务器作为宿主机，SSH连接后进行开发、部署和调试。容器化后端应用时选择使用docker-compose组建一个多容器（双容器，一个为Flask App容器另一个为MySQL容器）的Docker应用程序，两个容器之间可以相互调用。运行Python Flask程序的容器将容器内5000端口映射到宿主机5000端口，方便另外部署的前端页面调用后端API。装有MySQL的容器则将容器内的3306端口映射到3300端口，方便在外界有图形用户界面的计算机使用MySQL Workbench对数据库进行连接并管理。后端容器化部署构架见图5-3。

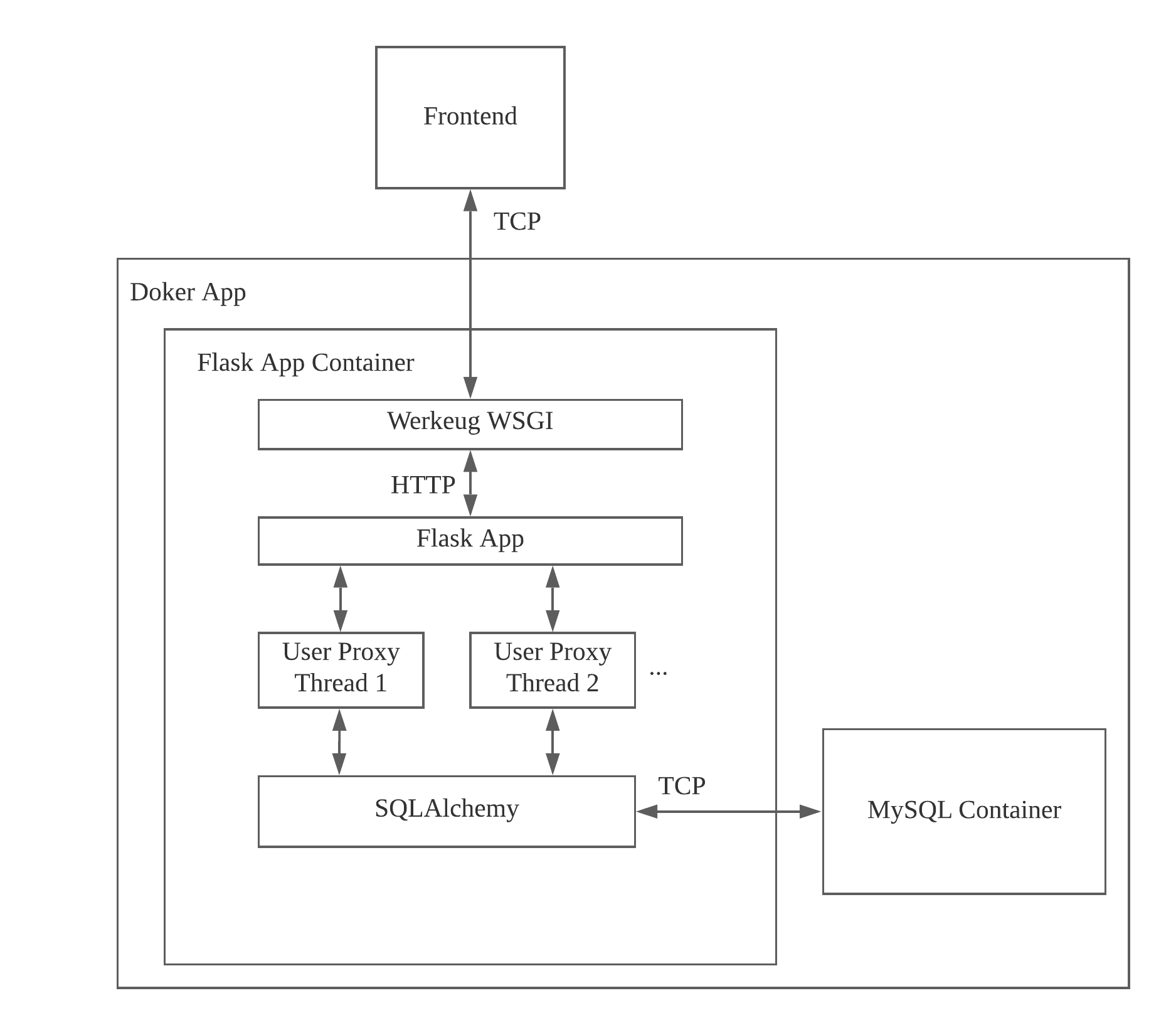


图5-3 后端架构

后端构架图解：

1. 宿主机（阿里云弹性服务器）安全组设置中允许入方向（Inbound）TCP到5000端口，从而使任意IP来源的请求都可以调用后端API。
2. Docker容器内5000端口映射到宿主机5000端口，访问到宿主机5000端口的TCP包可以抵达Docker容器内运行的Python Flask框架后端。
3. Flask依赖的WSGI，Werkeug接收TCP包将其拆包得出HTTP请求交给Flask App。同样出境的响应也经由WSGI封装传出。
4. Flask App接收到HTTP请求后判断是否已经存在用户代理（User Proxy），若有用户代理直接进入下一步，若无则创建用户代理线程为用户请求做后续相关的服务。由于可能会出现多线程读写同一个临界资源，编程实现需要实现互斥访问。
5. 每个用户代理线程会使用SQLAlchemy提供的MySQL数据库引擎创建与MySQL数据库的连接，创建数据库访问的Session，从而实现后续对数据库的增删查改。
6. 用户代理读写数据库，对数据进行审核，加工，筛选并组织成需要的格式。做出响应。
7. 响应流程大致如下：处理完数据存放在字典结构中返回，将数据序列化为JSON格式数据，再将其封装到flask.Response，最后由WSGI封装传出TCP报文。

5.2 界面设计

5.2.1 共享停车平台宣传页设计

共享停车平台网站宣传界面上方居中是根据网站名Shared Parking设计的图标，左上是社交媒体账号链接，右上则是登录注册选项。宣传界面主体是一个轮播的宣传图以及宣传文案。宣传主界面布局见图5-4。

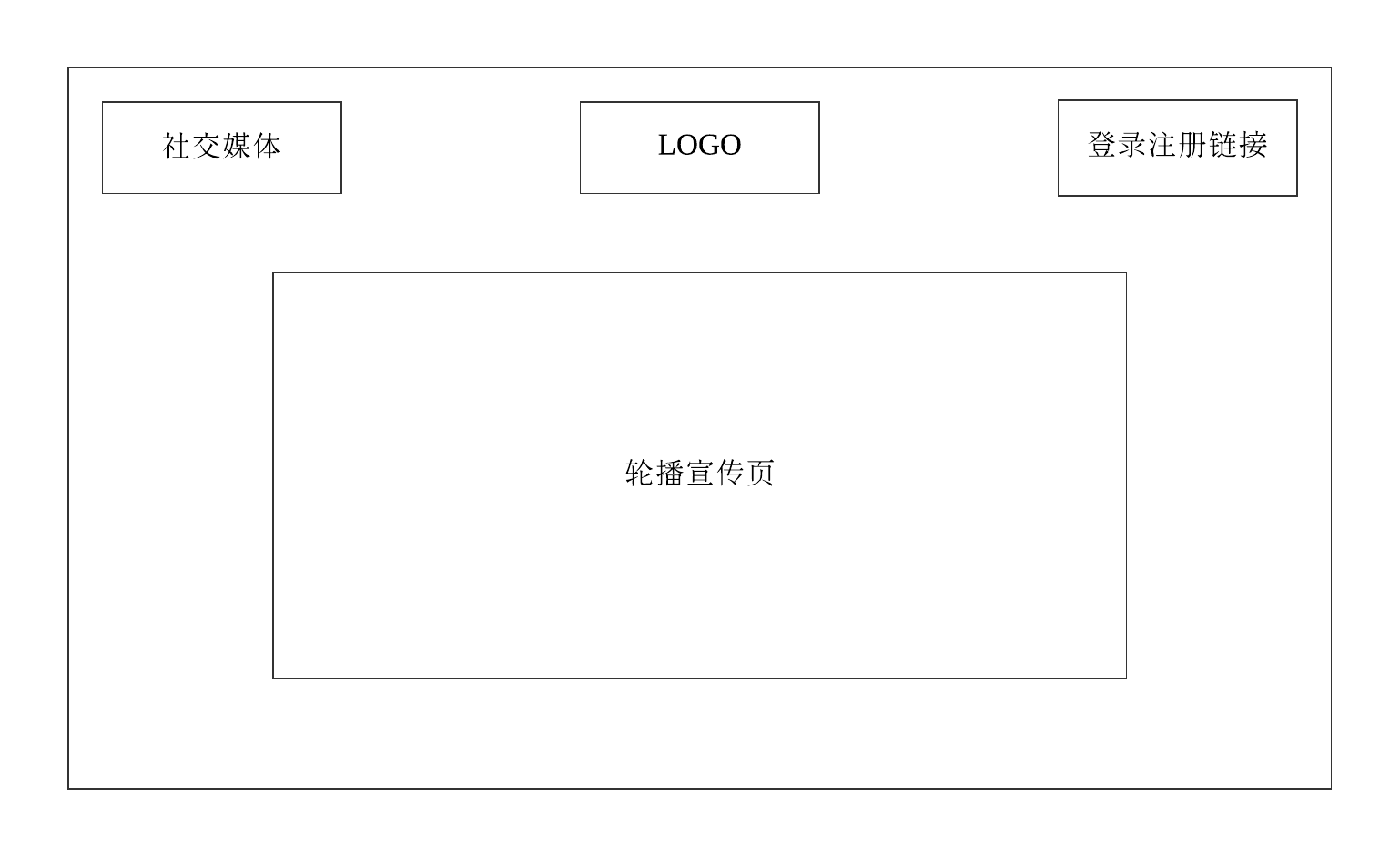


图5-4 宣传界面布局

5.2.2 共享停车平台登录注册设计

共享停车平台登录注册页面大体相同，上方是网站图标，居中设置输入框，下方一个提交按钮。背景做了渐变移动的效果。提示框会在页面上方居中弹出。登录界面示意图如5-5，注册界面5-6。



图5-5 共享停车平台登录页面布局



图5-6 共享停车平台注册页面布局

5.2.3 用户主界面设计

主界面（登录后自动跳转到的页面）设计布局如下：顶部为带有商标以及控制按钮的顶部横幅；左侧为可以隐藏或弹出的选项菜单，如图5-8；右侧为可以在查询车位时弹出的车位列表，如图5-9。主页面布局如图5-7。

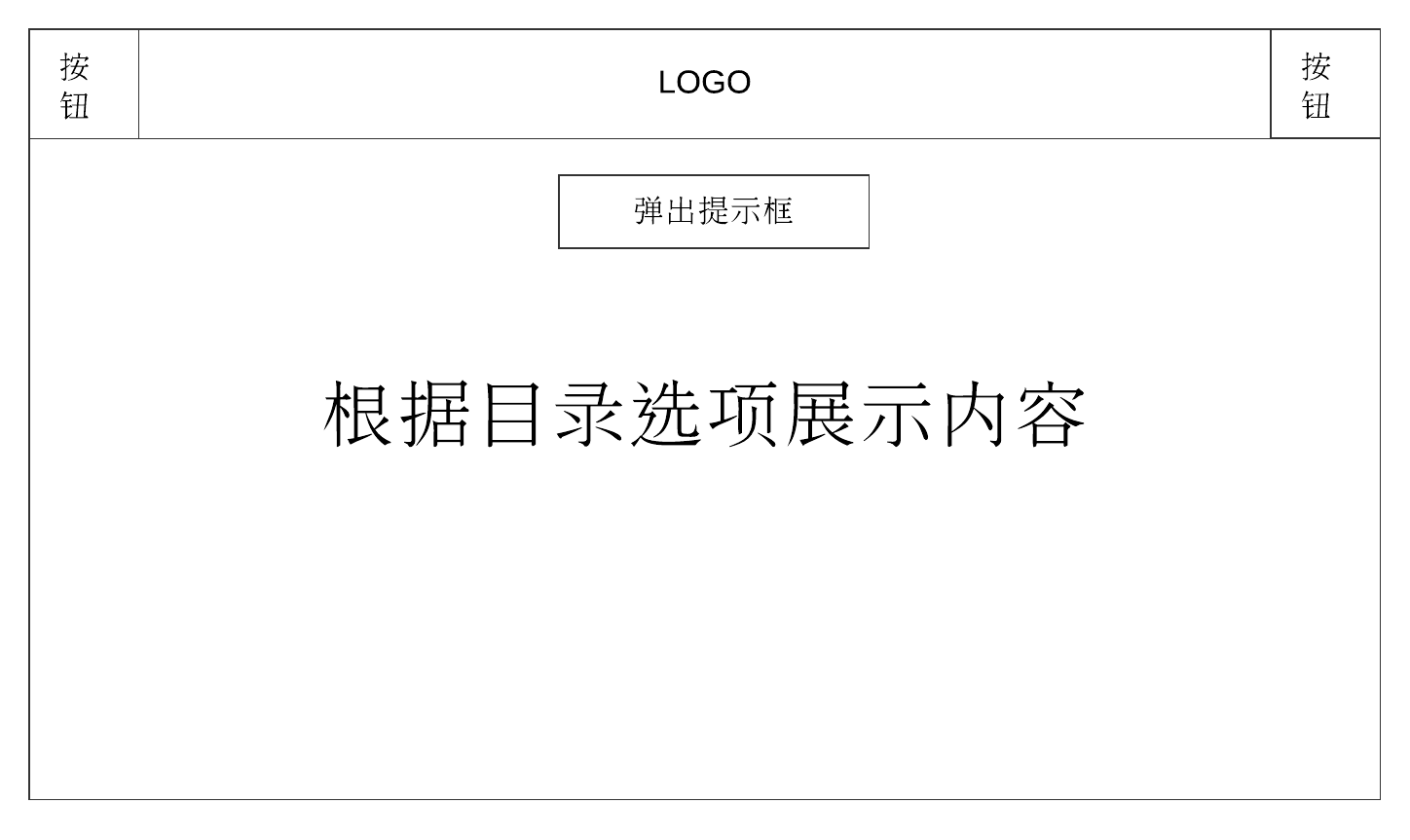


图5-7 主页面布局



图5-8 主页面左侧弹出时布局

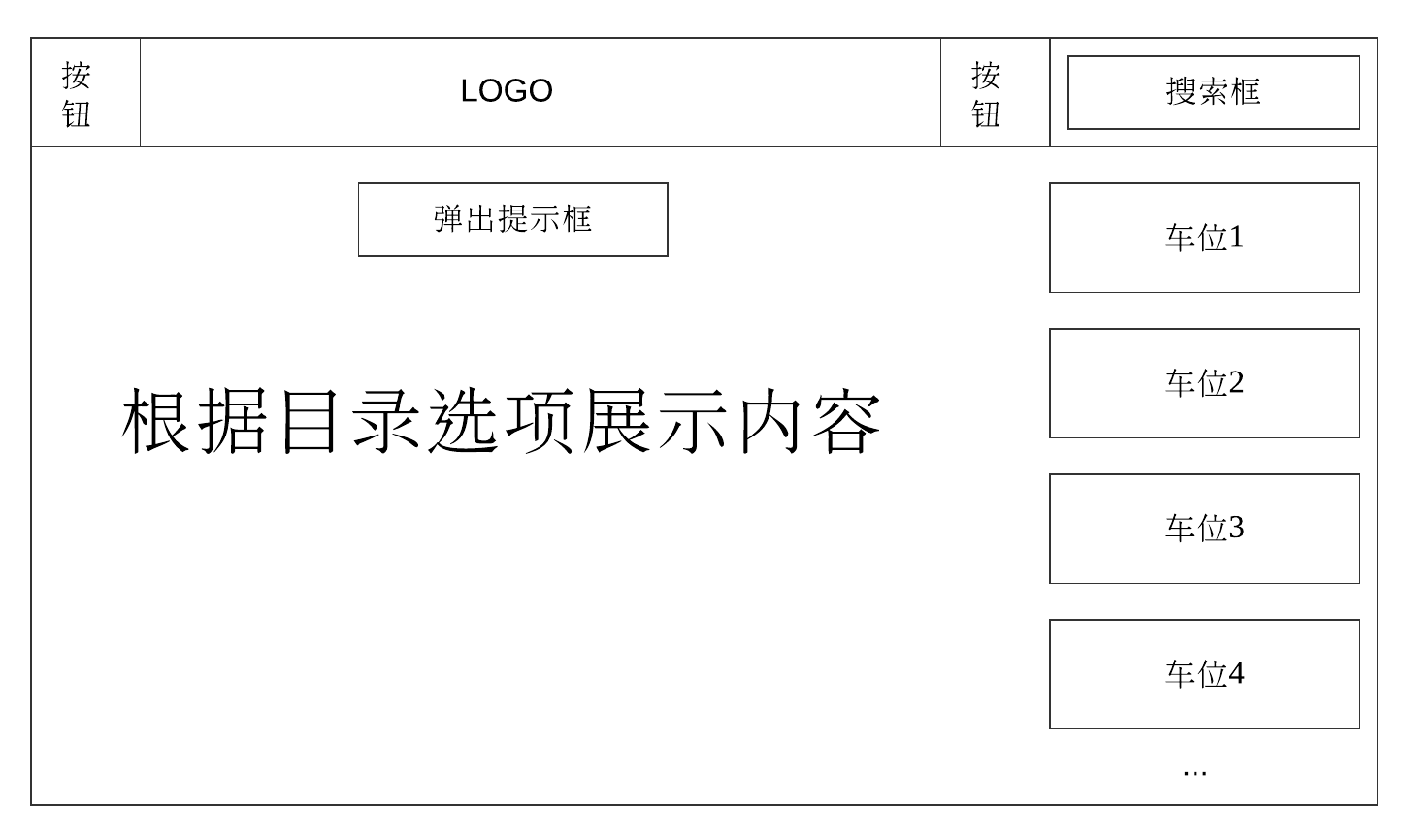
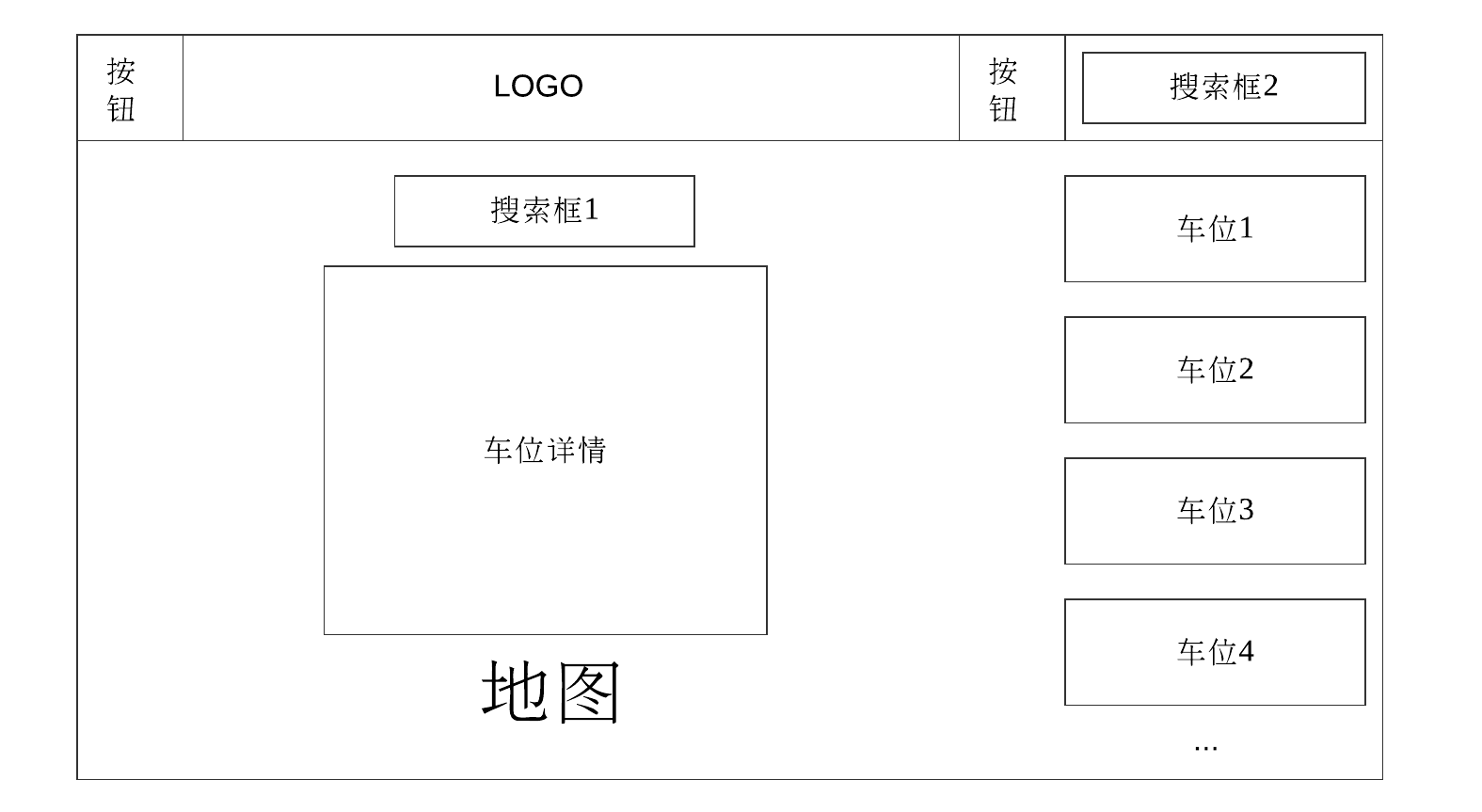


图5-9 主页面右侧弹出时布局

5.2.4 查询车位界面设计

当在左侧目录栏选择查询车位功能时，页面展示地图，搜索栏以及右侧车位列表。具体页面布局如图5-10。

图5-10 查询车位页面布局

5.2.5 订单管理界面设计

主界面左侧菜单栏选择订单管理后，页面内容展示框展示订单管理相关界面。布局如下图5-11。

图5-11 订单管理页面布局

5.2.6 车位管理界面设计

主界面左侧菜单栏选择车位管理，内容部分展示车位管理相关界面。概览如下图5-12；添加车位界面布局如图5-13；车位详细情况如图5-14。

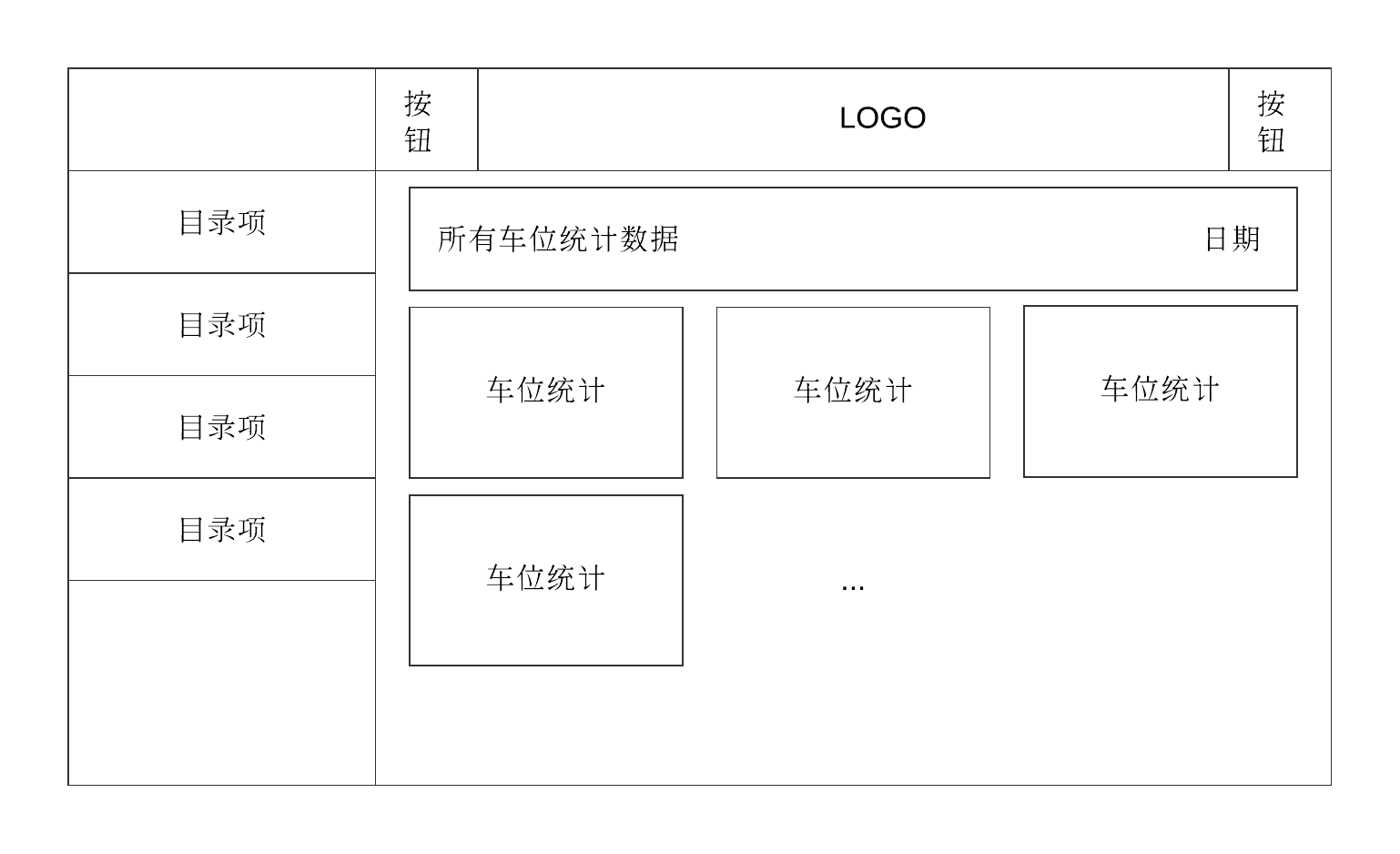


图5-12 车位管理页面布局

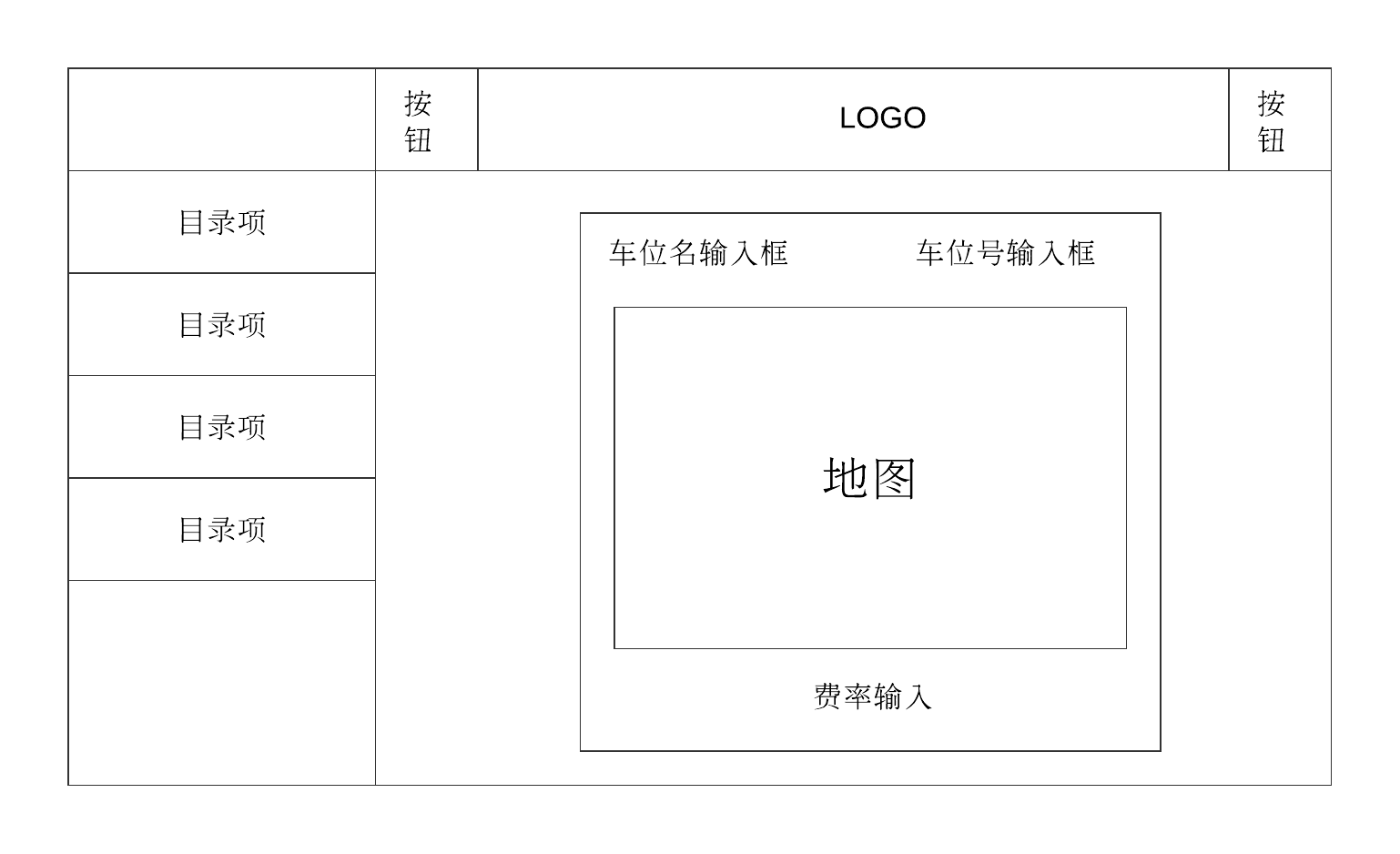


图5-13 车位管理-添加车位布局

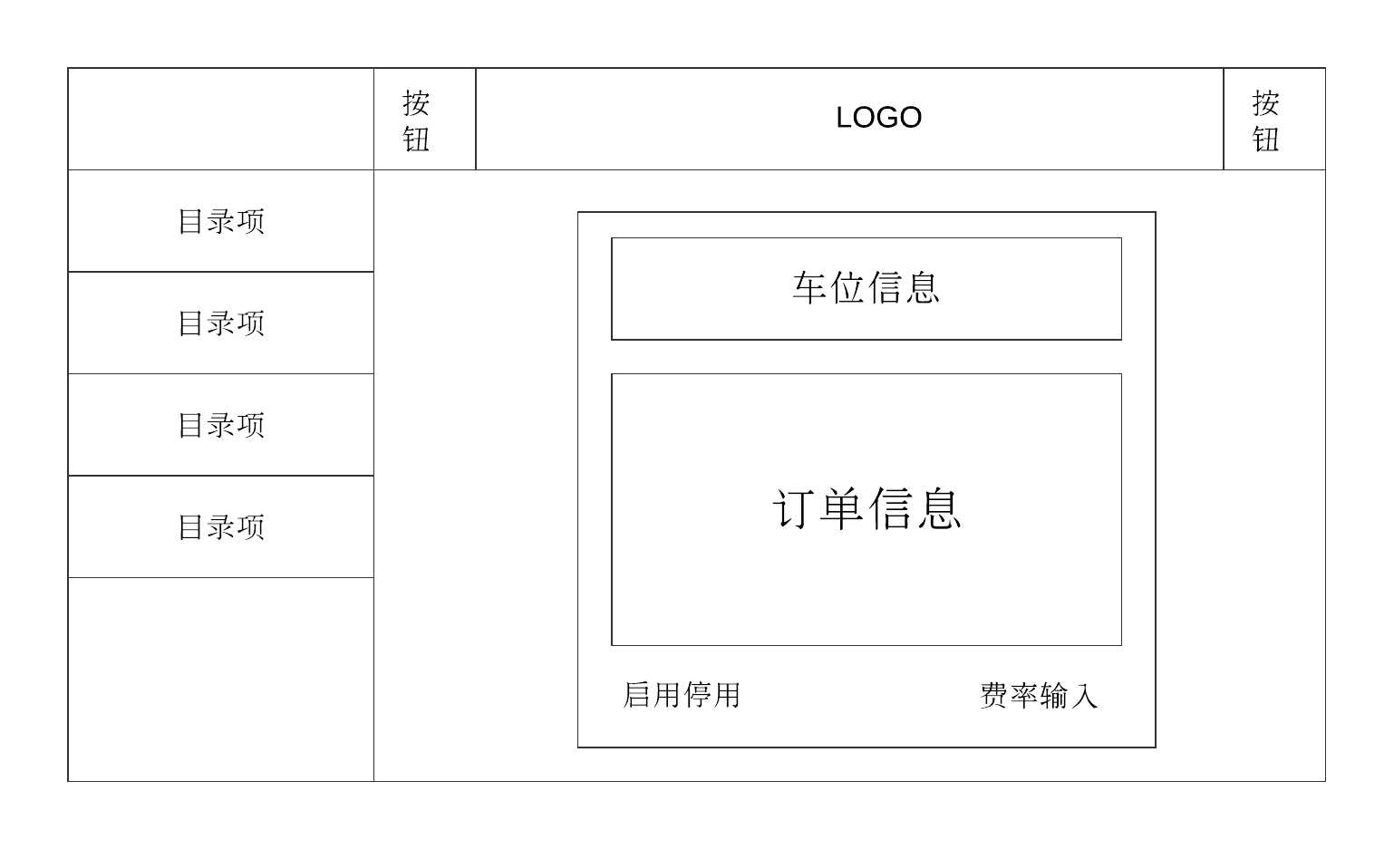


图5-14 车位管理-车位详情布局

5.2.7 账号理界面设计

主界面左侧菜单栏选择账号管理即可展示账号管理界面，账号管理界面设计如图5-15，可以方便用户查看基本账号信息，并对一些账号信息进行修改。

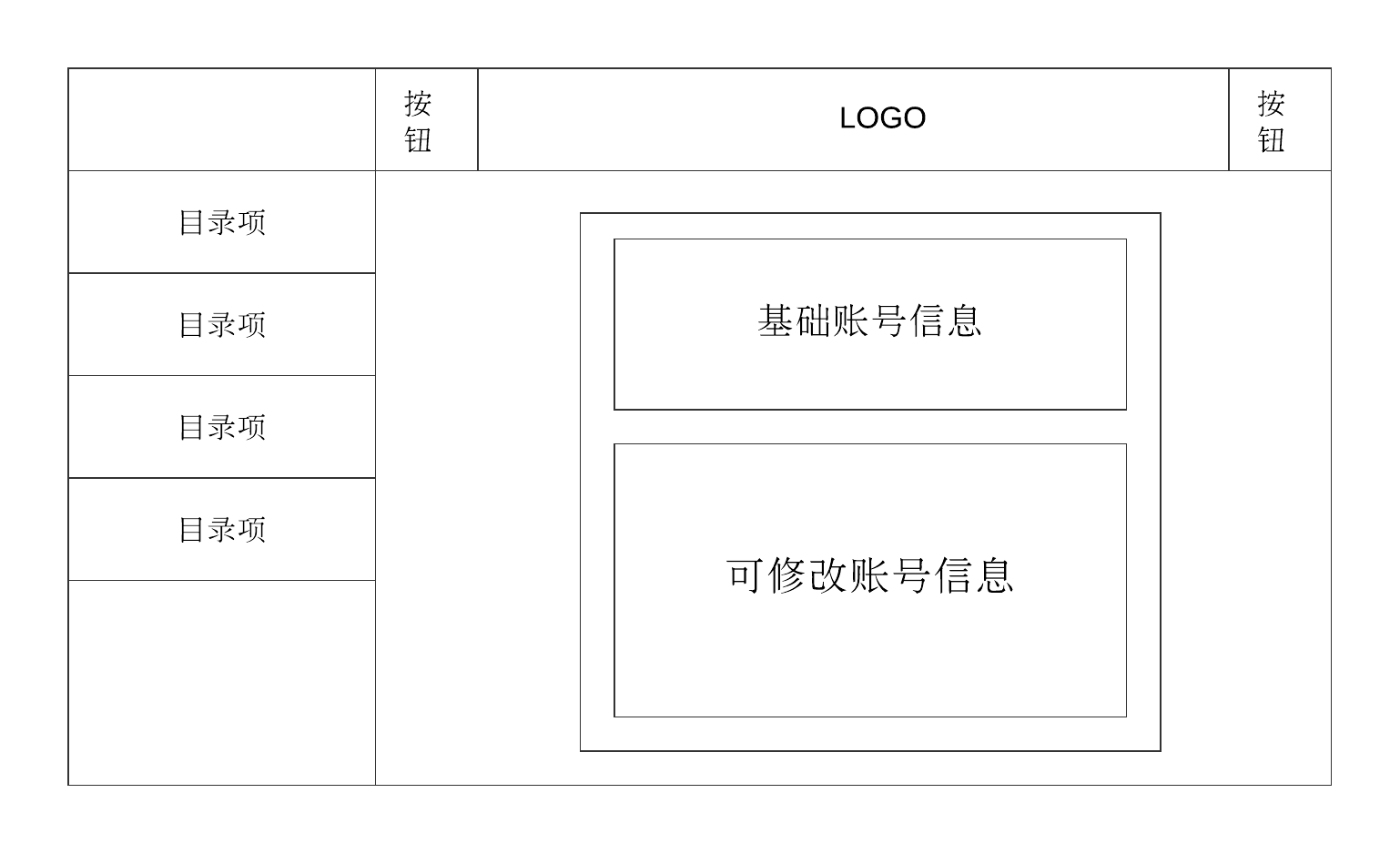


图5-15 账号管理页面布局

5.3 功能模块详细设计

5.3.1 账号登录以及注册详细设计

用户注册时当输入各项数据，手机号使用正则表达式对其长度以及内容限制。输入手机号完成后向后端发送请求检测手机号是否已被注册。下一步检查密码与输入的确认密码是否一致。各表单项都使用setCustomValidity设置不合法提示信息以及在提交前做出判断。注册成功后将手机号保存在SessionStorage中，页面跳转至登录界面，并将保存的手机号取出放在输入框中，这样用户不需要再次输入手机号。登录时使用MD5将密码哈希放在请求中传输，以防数据包被恶意监听从而原始密码泄密。登录以及注册后端对应的处理函数为UserProxy.register以及UserProxy.login。

5.3.2 按照车位名搜索功能详细设计

在搜索车位界面右侧弹出栏的输入框中输入关键词，点击搜索。右侧边栏Vue绑定的元素el获取搜索框中的输入值、浏览器获得的用户经纬度传到后端API（前端使用Axios像后端API提交请求）。后端收到查询词在数据库查询语句中使用LIKE关键字匹配车位名。返回出的车位列表依靠离定位的距离升序排序，返回给前端动态列表在页面上显示。接口调用函数为UserProxy.search\_pname。

5.3.3 位在目的地附近搜索车位功能详细设计

加载搜索车位界面后网页主体为地图，在中间搜索框输入目的地名称，并在弹出的地址列表选择确定的目的地，由于在搜索框元素在initMap时加入了addListener，选择后会将目的地的经纬度传到后端API。后端使用半正矢公式（Haversine Formular）计算球面上两坐标距离。返回距离目的地两公里内的停车位信息，并在地图上渲染出图标。图标的自定义通过创建新的google.maps.Marker实例时指定icon的url参数为自定义图片的路径即可实现。多个车位Marker放在一个JavaScript列表中方便遍历以及操作。

点击停车位图标将弹出infowindow展示车位大致信息，再点击infowindow则展示车位预定界面。

5.3.5 观看网课模块详细设计

用户登录完成后在查看网课页面点击“立即加入”，调用jumpPlayVideo(String userId,String courseId)方法，再通过service和dao查询出课程的章节列表，用户选择章节，异步调用getInchapterVideoList(String courseId,String chapterId)方法得到章节内的视频列表，用户选择播放的视频，异步调用userPlayVideo(String courseId,String chapterId,String inChapterId)获取视频内容、收藏数量和评论内容。

5.3.6 收藏模块详细设计

视频播放页加载出视频信息的同时，调用service层的whetherUserCollectVideo(String userId,String videoId)和countVideoCollectNum(String video)查询用户是否已经收藏该视频和该视频已经被收藏的次数，如果用户已经收藏该视频，则收藏按钮为实心，反之为空心。用户点击收藏按钮就可实现对正在观看的视频的收藏或取消收藏。用户点击“我的收藏”，调用myCollect(String userId)方法查询出用户收藏的视频并分页显示，点击视频即可跳转到播放界面。

5.3.7 评论功能模块详细设计

在视频播放页面，视频加载时，会调用service层的findComment(String userId,String videoId)将本用户发表的评论加载出来，在前端通过js方法setComment(point,comment,time)显示出用户的评价。用户新发表或修改评价是调用createComment()实现保存评价。点击导航栏“我的评论”按钮，在新页面会显示出本用户发表的所有评价，用户可以调用deleteComment(String comment)删除不合理的评价。

5.3.8 账户设置功能模块详细设计

账户设置包括头像修改、兴趣修改、邮箱修改、密码修改和邮件功能设置。用户在修改界面选择图片文件，可以实现头像预览，点击上传按钮调用UploadFileController里面的userAccountImg(MultipartFile multipartFile,String userId)方法实现头像上传到服务器功能，并通过service层调用dao层的updataUserImg(String userId,String imageAddr)更新数据库里用户头像地址。

修改邮箱操作如下，用户在输入框内输入新的电子邮箱，点击获取验证码，调用SendEmailController里面的verificationCode(String emailAddr)方法，随机生成6位数字验证码，再调用SendEmail类的静态方法sendMail(String emailAddr,String msg,String title)向修改的邮箱发送验证码。用户收到邮件后，在修改页面填入验证码，点击调用checkEmailCode(String code)方法验证码验证码是否正确，若正确则通过service调用dao层modifyUserEmail(String userId,String email)修改用户的电子邮箱。

修改密码时，用户需要输入账户登录的旧密码和两次新设置的密码，点击修改调用modifyPassword(String userId,String oldPass,String newPass)方法，先验证旧密码是否正确，若正确则修改用户密码，返回后让用户重新登陆；若错误则提示用户，密码错误，修改失败。

进入设置兴趣页面时，会自动显示用户已设置的兴趣，用户可以点击下面系统预设的待选项，将内容填的输入框，点击修改，依次调用controller、service和dao更新个人兴趣。

邮件功能设置主要为用户设置自己的邮箱是否接手课程更新、机构、平台的邮件提醒，默认全部接收，用户可以修改复选框状态，点击保存，调用modifyEmailFun()更新邮件功能设置。

5.3.9 反馈模块详细设计

在反馈页面，用户需要填入用户名、密码、反馈内容和电子邮箱，其中密码为选填，点击提交后，调用FeedBackController中的create(String username,String pwd,String feedback,String email)方法，先用验证用户名密码，若验证通过，该条反馈标记为已验证，反义为未验证，再依次调用FeedBackServiceImpl和IFeedBackDao的create(FeedBackInfo feedBack)方法将反馈存入数据库。

5.3.10 课程管理功能模块详细设计

课程管理功能主要分为课程的创建和修改，视频的上传和修改。课程创建需要在表单里填写完整课程信息，调用CourseController里面的createCourse()方法上传视频封面，保存视频信息到数据库内的视频表。修改视频需要先调用institutionModifyCourseInfo(String courseId)查询出该课程的信息，通过返回ModelAndView在JSP页面使用EL表达式显示出来，机构用户修改完后，点击保存调用modifyCourseInfo(CourseInfo courseInfo)方法，先使用isEmpty()判断是否修改课程封面，再将新的课程信息更新到课程表内。视频的上传和修改与课程类似，调用createVideo()上传视频、保存信息到视频表内；修改时调用institutionModifyVideoInfo(String videoId)先查询出视频信息，修改完成后调用modifyVideoInfo(VideoInfo videoInfo)完成视频的更新和信息的保存。

5.3.11 用户评论管理功能模块详细设计

机构用户进入用户评论管理页面时调用JumpController里面的institutionMagComment(String institutionId)查询出机构创建的课程内所有评价，并分页显示，若发现不当评论时，可以点击删除，调用Controller层的institutionDelete(String commentId)方法删除用户的评论。

5.3.12 账户信息变更功能模块详细设计

机构用户在账户信息变更页面可以修改机构名称、资质照片和机构简介，点击保存调用InstitutionAccountController里面的modifyInstitutionInfo()上传新的资质照片和更新机构信息。此时，机构账户变为功能受限状态，只能访问机构主页和账户信息变更页面，待管理员审核通过后方可恢复正常；若管理员审核未通过，机构用户可以进入信息变更页面修改信息，再次提交申请，直至审核通过。

5.3.13 机构数据中心模块详细设计

点击进入机构数据中心，调用institutionDateCharts(String institutionId)，在依次调用InstitutionDataService和IInstitutionDataDao的方法查询出各项数据，通过ModelAndView返回至数据展示页面，将整理好的数据放入echarts内，生成相应的展示图标。

5.3.14 首页轮播管理模块详细设计

点击轮播图管理，调用AdminJumpController的carouselMag()的方法，查询并分页显示已有的轮播图信息，管理员可以点击按钮调用display(String carouselId)或hide(String carouselId)实现对轮播图的展示和隐藏。在此页面里，管理员也可以上传新的轮播图片，填写完整轮播图信息，选择对应的图片，点击上传按钮，调用CarouselController的upload()方法，上传轮播图片至服务器，同时将信息写入轮播图表。

5.3.15 用户管理模块详细设计

在用户管理页面里，可以分别查看机构账户信息和普通用户的账户信息，在此模块内，调用open(String id)和stop(String id)可以对两种账户进行停用和解封的操作。对机构账户注册或修改申请的审批工作，需要移步至事务审批模块操作。

5.3.16 事务审批模块详细设计

此模块主要是对机构注册和信息变更申请做出审批，通过institutionCheck()查询出所有待审批的账户信息，使用表格的形式分页展示，这些账户状态分为“注册待审核”、“变更待审核”，调用institutionPass(String institutionId)通过审核，机构账户恢复正常，或者institutionRefuse(String instutitonId)拒绝申请或修改，机构账户变为“已终结，待重申”。

5.3.17 反馈处理模块详细设计

调用feedBackHandle()查询出所有没有回复的用户反馈，管理可以选择点击忽略，调用ignore(String feedbackId)方法，忽略此条用户反馈；也可以点击回复，输入回复的内容，点击发送，调用handle(String feedBackId,String email,String reFeedBack)方法来给用户发送回复邮件，同时将回复内容存入反馈表。

5.3.18 后台数据中心模块详细设计

点击进入后台数据中心，调用adminDateCharts()，在依次调用adminDataService和adminDataDao的方法查询出各项数据，通过ModelAndView返回至数据展示页面，将整理好的数据放入echarts内，生成相应的展示图标。

# 6 系统实现

6.1 注册登录功能实现

进入网页后，点击“点击登录/注册”，跳出注册登录弹框，用户根据自己实际情况选择注册或登录，登录过程中会获取用户的IP地址和登录地址，该过程用户不可知，登录注册界面如图6-1所示。

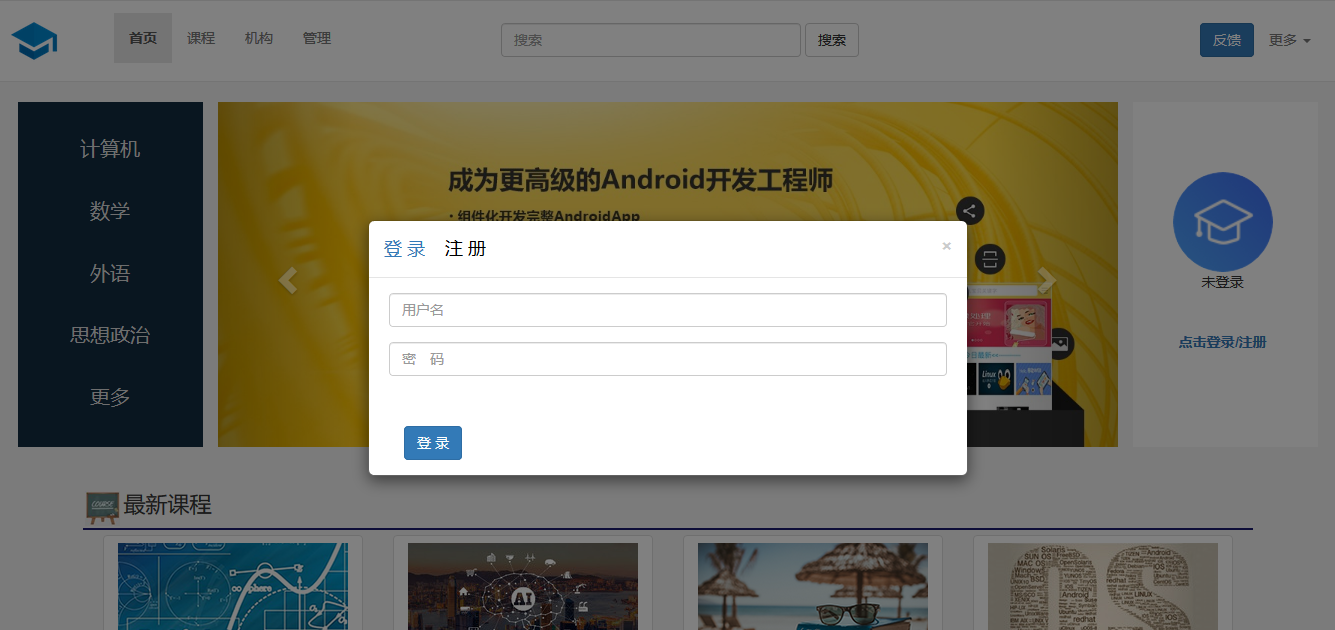


图6-1 登录注册界面

访问ip138.com和调用百度地图API获取用户IP和登录地址信息实现代码如下：

url = new URL("http://2020.ip138.com/ic.asp");

urlconn = url.openConnection();

调用百度API获得城市代码如下：

String ak="fnQ7V6EgNp2FWaYwNllRTd4BnpnAwuOy";

String url="http://api.map.baidu.com/location/ip?ak="+ak+"&ip="+ip+"&coor=bd09ll";

InputStream is = new URL(url).openStream();

BufferedReader rd = new BufferedReader(new InputStreamReader(is, Charset.forName("UTF-8")));

String jsonText = readAll(rd);

JSONObject json = JSONObject.parseObject(jsonText);

JSONObject obj = (JSONObject) ((JSONObject) json.get("content")).get("address\_detail");

String province = obj.getString("province");

String city=obj.getString("city");

6.2 主页课程推荐功能实现

进入系统主页，随机挑选管理员设置的5张轮播图片（如图6-2所示），此外自动给用户推荐最新课程（如图6-3所示）、最热课程和个性推荐，点击轮播图片或这栅格系统显示的课程，即可进入对应的课程简介页面。



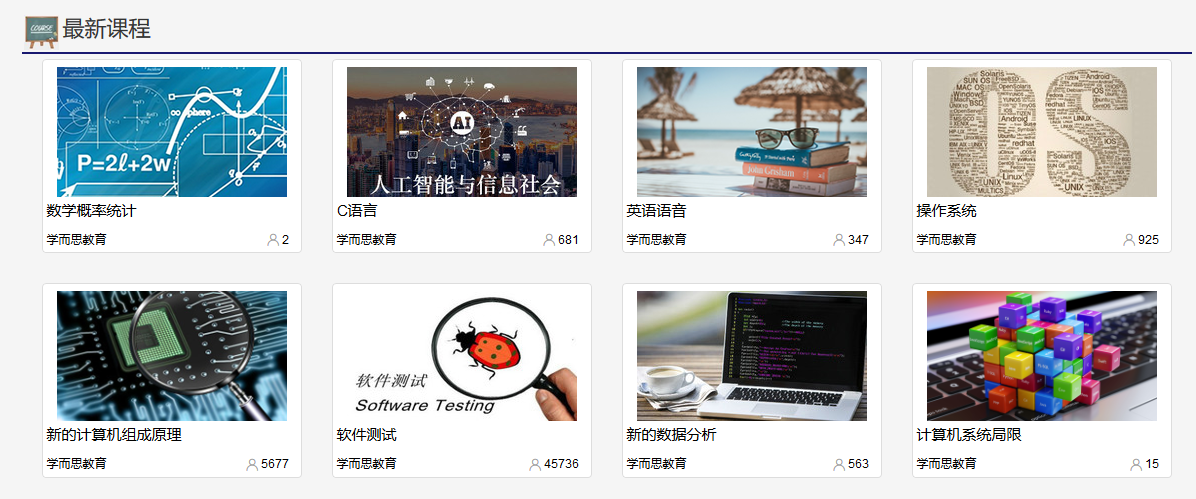
图6-2 首页轮播图界面 

图6-3 首页课程推荐界面

获取轮播图代码如下：

@RequestMapping("/getCarousel")

public @ResponseBody Map<String, List<CarouselInfo>> getCarousel(){

try{

List<CarouselInfo> homePageCarouselInfo = iAdminCarouselMagService.getHomePageCarouselInfo();

for(int i=0;i<homePageCarouselInfo.size();i++){

System.out.println(homePageCarouselInfo.get(i));

}

map.put("homePageCarouselInfo",homePageCarouselInfo);

}catch (Exception e){

System.out.println(e);

}

return map;

}

生成个性推荐的代码如下：

String[] interestSplit = interest.split(",");int len=interestSplit.length;

for(int i=0;i<len;i++){

recoUserHomePageCourse= iCourseService.findRecoUserHomePageCourse("%"+interestSplit[i]+"%");

recommendUserHomePageCourse.addAll(recoUserHomePageCourse);

}

homePageCourseSet.addAll(recommendUserHomePageCourse);

recommendUserHomePageCourse.clear()；

while (homePageCourseSet.size()<8){

recommendUserHomePageCourse=iCourseService.findRandUserHomePageCourse();

homePageCourseSet.addAll(recommendUserHomePageCourse);

recommendUserHomePageCourse.clear();  
 }

int count=0;

for(PreviewCourseInfo previewCourseInfo:homePageCourseSet){

count++;recommendUserHomePageCourse.add(previewCourseInfo);

if(count==8){break；}

}

在代码实现上，获取最新课程信息、最热课程信息的代码与个性推荐代码类似，故不再重复赘述。

6.3 课程搜索功能实现

在系统主页的搜索框中输入想要搜索的课程关键词（如图6-4所示），按下回车键或者点击搜索，即可分页显示搜索结果（如图6-5所示），点击可进入课程信息介绍页面进行查看。



图6-4 搜索内容展示界面

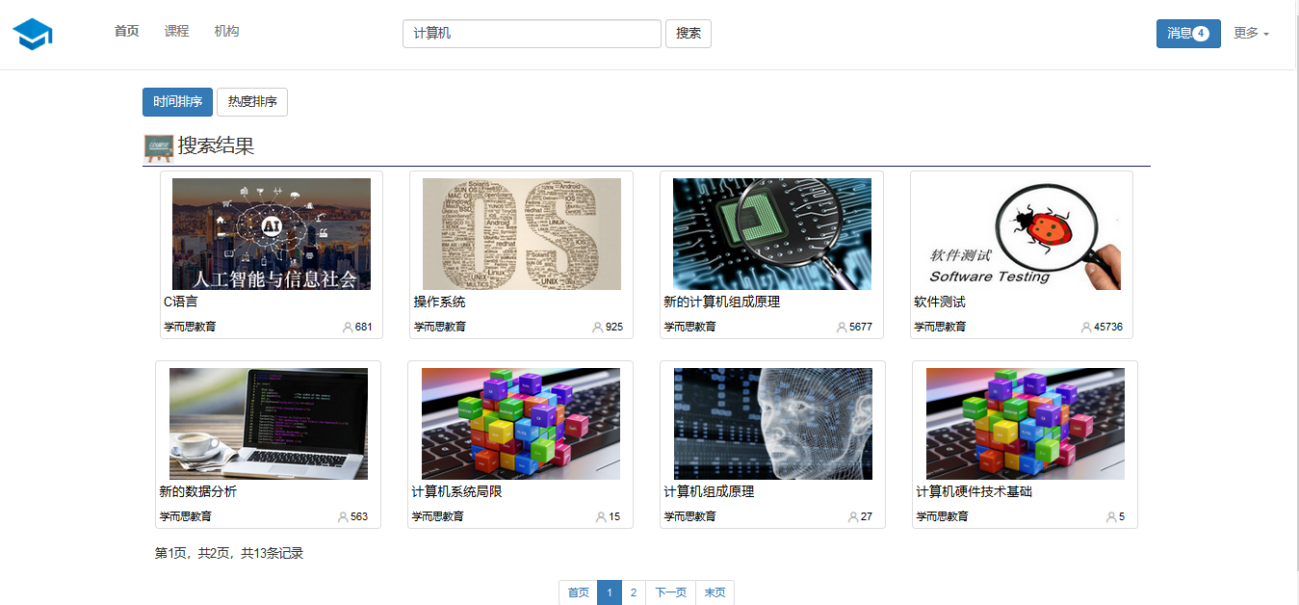


图6-5 搜索结果展示界面

搜索功能前端代码：

$("#search\_btn").click(function () {

if(searchText!=""){

var url="${pageContext.request.contextPath}/search/sortByCreateTime?searchKey="+searchText;

$("#searchA").attr("href",url);

var el=document.getElementById('searchA');

el.click();

}

});

对应后台代码如下：

（1）使用mybatis的注解写的sql语句如下：

@Select("select courseId,courseName,courseImageAddr,uploaderName,studentNum from course\_info where kindKey like #{searchKey} or courseName like #{searchKey} Order By createTime Desc ")

public List<PreviewCourseInfo> searchOutComeSortByTime(String searchKey);

（2）Java代码如下：

char[] searchKeyChar = searchKey.toCharArray();

String searchKey2="%";

for(int i=0;i<searchKeyChar.length;i++){

searchKey2=searchKey2+searchKeyChar[i]+"%"; }

List<PreviewCourseInfo> previewCourseInfoList = iCourseService.searchOutComeSortByTime(page, size, searchKey2);

PageInfo previewCourseInfoListPageInfo=new PageInfo(previewCourseInfoList);

6.4 查看课程功能实现

在课程信息查看界面先看到课程的重要信息（如图6-6所示），接下来是机构的简介、课程简介和课程目录信息（如图6-7所示），最下面是分页显示的用户评价（如图6-8所示）。



图6-6 查看课程界面



图6-7 课程和机构简介界面



图6-8 课程所有评价界面

获取课程信息代码如下所示：

CourseInfo courseInfo = iCourseService.findCourseByCourseId(courseId);

List<ChapterInfo> chapterList = iCourseService.findChapterBeanByCourseId(courseId);

List<String> inChapterVideoList=null;

for(int i=1;i<=chapterList.size();i++){

inChapterVideoList=iVideoService.findInChapterVideoList(courseId,i);

}

获取评论后台代码如下：

List<UserCommentInfo> userCommentInfoList=iCommentService.findCourseAllComment(page,size,courseId);

UserCommentInfo userCommentInfo=null;

for(int i=0;i<userCommentInfoList.size();i++){

userCommentInfo=userCommentInfoList.get(i);

userCommentInfo.setCreateTime(GetTime.cutDateTime(userCommentInfo.getCreateTime()));

userCommentInfoList.set(i,userCommentInfo);

totle=totle+userCommentInfo.getPoint();

}

double avg=totle/userCommentInfoList.size();

int avg\_int= (int)Math.round(avg);

PageInfo userCommentInfoListPageInfo = new PageInfo(userCommentInfoList);

6.5 观看网课功能实现

用户在课程信息查看页面上点击开始学习，即可进入网课播放页面，在选择往章节和章节内视频列表后，即可显示该视频的信息（如图6-9所示），点击即可播放视频。



图6-9 网课播放界面

加载视频前端ajax代码如下：

$.ajax({

url:"${pageContext.request.contextPath}/video/userPlayVideo",

data:{"courseId":'${courseInfo.courseId}',"chapterId":chapterId,"inChapterId":inChapterVideoId},

type:"post", dataType:"json",async:false, traditional : true

});

后端代码如下：

（1）获取视频信息并保存播放记录：

VideoInfo videoInfo=iVideoService.findVideoBean (courseId,chapterId,inChapterId);

PlayVideoInfo playVideoInfo = new PlayVideoInfo(playId, userId, courseId, videoId, playTime);

iVideoService.saveUserPlayInfo(playVideoInfo);

（2）获取收藏和评论信息

CommentInfo commentInfo=iCommentService.findCommentInfoById(userId,videoId);

if(commentInfo!=null){

commentInfo.setCreateTime(GetTime.cutDateTime(commentInfo.getCreateTime()));

}

int collectNum = iUserCollectVideoService.countVideoCollect (videoId);

int state=iUserCollectVideoService.whetherUserCollect (userId,videoId());

6.6 收藏功能实现

用户在“我的收藏”里面可以看到自己收藏的所有视频，视频预览信息包括所属机构、视频标题、收藏时间和累计播放量（如图6-10所示）。

图6-10 用户我的收藏界面

展示收藏视频的后端代码如下：

List<PreviewVideoInfo> userCollectVideo = iVideoService.ollectVideo(page,size,userId);

for(int i=0;i<userCollectPreviewVideoInfo.size();i++){

videoInfo= userCollectVideo.get(i);

videoInfo.setCollectTime(GetTime.cutDateTime(videoInfo.getTime()));

userCollectVideo.set(i,previewVideoInfo);

}

PageInfo userCollectVideoPageInfo=new PageInfo(userCollectPreviewVideoInfo);

mv.addObject("userCollectVideoPageInfo ", userCollectVideoPageInfo);

6.7 评论功能实现

用户在视频播放界面可以发表或修改评论，在“我的评论”里面可以查看自己发表的所有评论，用户可以删除自己发表的评论（如图6-11所示）。



图6-11 我的评论界面

前端删除按钮jquery代码如下：

$(".delete\_btn").click(function(){

var id=$(this).attr("id");

$.ajax({

url:"${pageContext.request.contextPath}/comment/delete",

dataType:"json",data:{"commentId":id}, type:"post",

success:function (data) {

if(data=="1"){

alert("评论删除成功！");

location.replace(location);

}

},

error:function (data) {

alert("删除评论发生异常！");

}

});

});

后台代码如下：

List<MyCommentInfo> userComment=iCommentService.findUserComment(page,size,userId);

List<Integer> pointList=iCommentService.findCommentPoint(userId);

double totle=0;

for(int i=0;i< pointList.size();i++){

totle=totle+ pointList.get(i);}

double avg=totle/ pointList.size();

int avg\_int= (int)Math.round(avg);

6.8 账户设置功能实现

点击账户设置，显示账户信息,用户可以修改电子邮箱和头像（如图6-12所示）。在邮件设置里面，用户可以自由设置电子邮箱功能（如图6-13所示）。兴趣设置和修改密码与其他两个界面类似，故不再叙述。



图6-12 修改信息界面

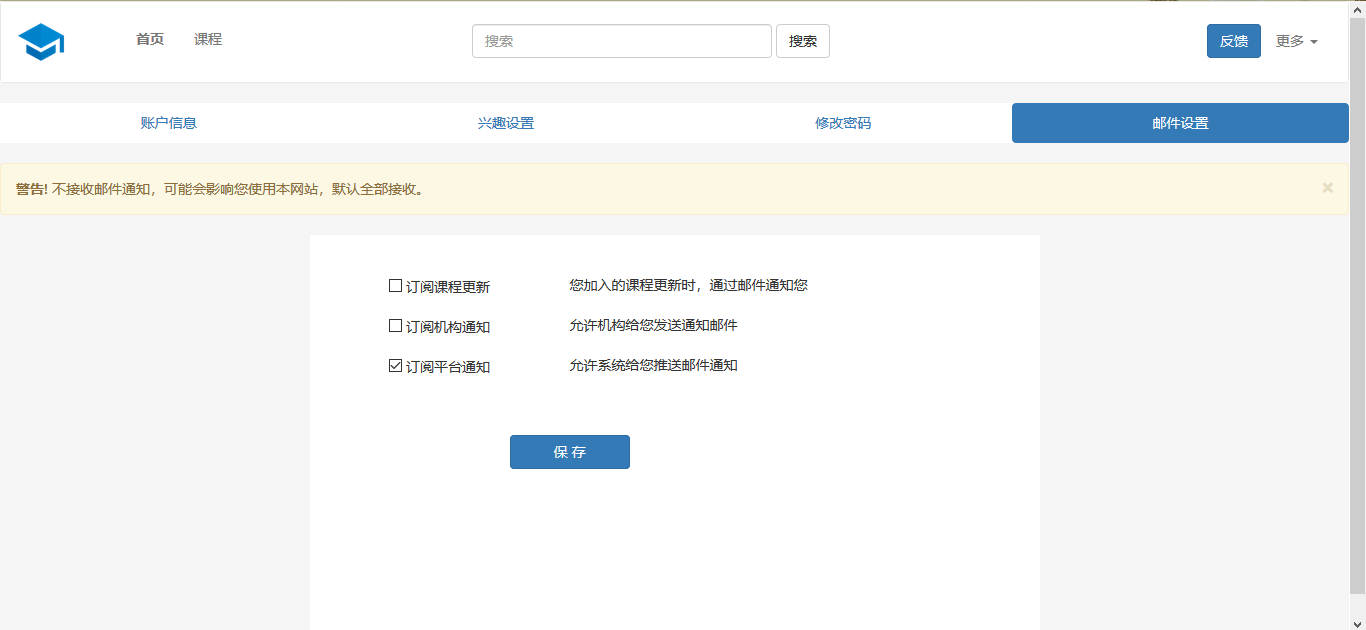


图6-13 邮件功能设置界面

发送邮件的代码如下：

Authenticator authenticator = new Authenticator() {

@Override

protected PasswordAuthentication getPasswordAuthentication() {

String userName = props.getProperty("mail.user");

String password = props.getProperty("mail.password");

PasswordAuthentication passwordAuthentication = new PasswordAuthentication(userName, password);

return passwordAuthentication;

}

};

Session mailSession = Session.getInstance(props, authenticator);

MimeMessage message = new MimeMessage(mailSession);

String username = props.getProperty("mail.user");

InternetAddress form = new InternetAddress(username);message.setFrom(form);

InternetAddress toAddress = new InternetAddress(to);

message.setRecipient(Message.RecipientType.TO, toAddress);

message.setSubject(title);

Transport.send(message);

6.9 反馈功能实现

任意用户界面的右上角都设置了反馈界面的入口，点击进入反馈界面，完整填写表单信息，然后提交用户反馈（如图6-14所示）。



图6-14 用户反馈界面

提交反馈后台代码如下：

@RequestMapping("/create")

@ResponseBody

public String create( String username, String password,String text\_feedback,String email) {

String feedBackId= GetOnlyID.getId();

int login = iUserAccountService.login(username, password);

int state=0;

String time= GetTime.getDateTime();

FeedBackInfo feedBackInfo;

if(login==1){

feedBackInfo=new FeedBackInfo(feedBackId,username,email,text\_feedback,time,1,"");

}else{

feedBackInfo=new FeedBackInfo(feedBackId,username,email,text\_feedback,time,2,"");

}

iFeedBackService.createFeedBack(feedBackInfo);

return "1";

}

6.10 评论功能实现

机构用户在查看课程界面可以分页查看课程信息（如图6-15所示），点击修改，即可进入到课程信息修改的页面（如图6-16所示），在此页面内可以修改对应的课程信息；点击查看按钮，可以查看课程内视频的信息，也可以修改并保存视频信息（如图6-17所示）。



图6-15 查看课程界面



图6-16 修改课程界面



图6-17 查看视频信息界面

查看所有课程信息代码如下：

@RequestMapping("/institutionDisplayCourseList")

public ModelAndView findAllCourse(int page,int size,String institutionId) {

List<CourseInfo> courseList=iCourseService.findAllCourse(page,size,institutionId);

PageInfo pageInfo = new PageInfo(courseList);

ModelAndView mv = new ModelAndView();

mv.addObject("pageInfo", pageInfo);

mv.setViewName("institutionDisplayAllCourse");

return mv;

}

上传课程封面的后台代码如下：

String actualP = request.getSession().getServletContext().getRealPath("/courseCoverImages/");

File file=new File(actualP);

if(!file.exists()){

file.mkdirs();

}

String originalFilename=multipartFile.getOriginalFilename();

String imgName=GetOnlyID.getFileName(originalFilename);

courseImageAddr="courseCoverImages/"+imgName;

multipartFile.transferTo(new File(realPath,imgName))；

6.11 用户评价管理功能实现

机构在用户评价管理页面，可以分页查看用户在其开设的课程下发表的所有评价，如若发现有不合理的评价，机构用户有权删除评价记录（如图6-18所示），其代码与用户的“我的评价”相似，故再次不在粘贴代码。

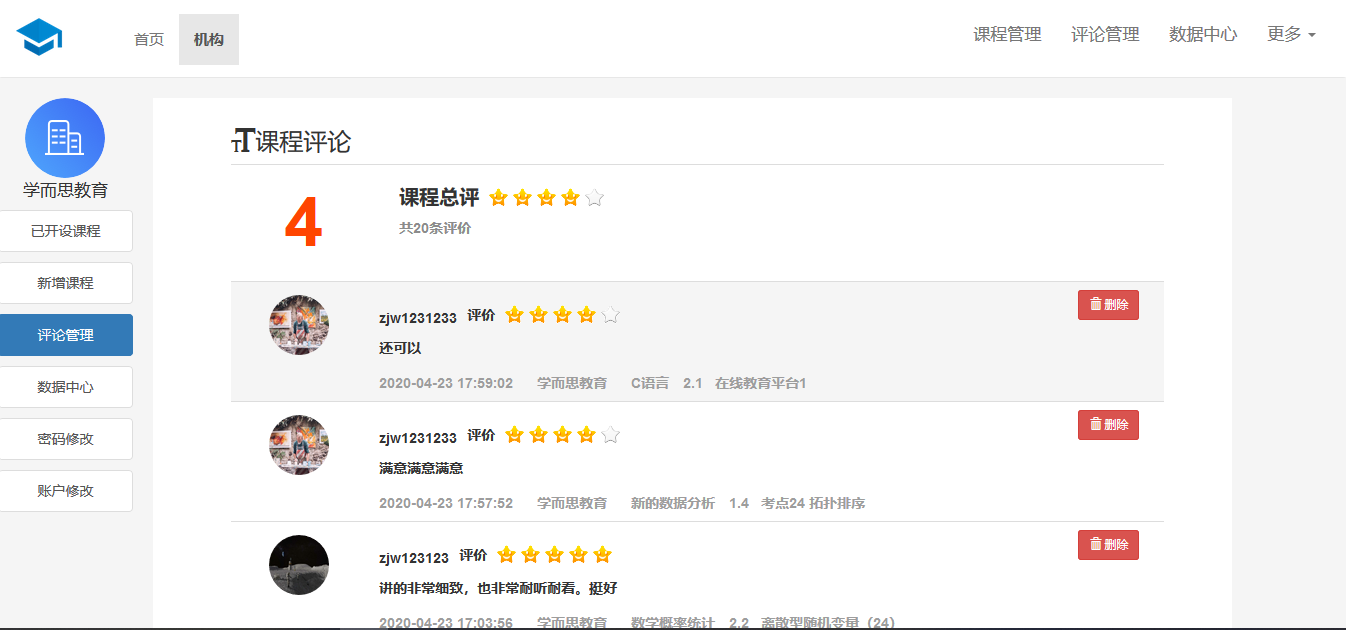


图6-18 用户评价管理界面

6.12 账户信息变更功能实现

机构用户在账户信息变更页面可以修改机构名称、上传新的资质证明照片和修改账户简介（如图6-19所示），点击保存即进入管理员审批流程，需要等管理员审批通过方可正常。



图6-19 机构账户信息变更界面

上传表单的前台ajax代码如下：

var form = document.forms.namedItem("institutionInfoForm");

var formData = new FormData(form);

$.ajax({

url: "${pageContext.request.contextPath}/institutionAccount/modifyInstitutionInfo",

type: "post",data: formData ,

success: function (data) {

if(data=="1") {

var el = document.getElementById('homePage'); el.click();

}

},

error:function (data) {alert("服务器端未知异常，请重试！"); }

});

6.13 机构数据中心功能实现

机构用户点击进入机构数据中心，页面上方将会显示参课人次、播放次数、评论数、收藏数的历史总数居和昨日数据，下方通过柱状图的形式展示进七日四中数据的变化情况（如图6-20示），方便机构用户掌握自己课程播放情况，以便及时调整课程类型。

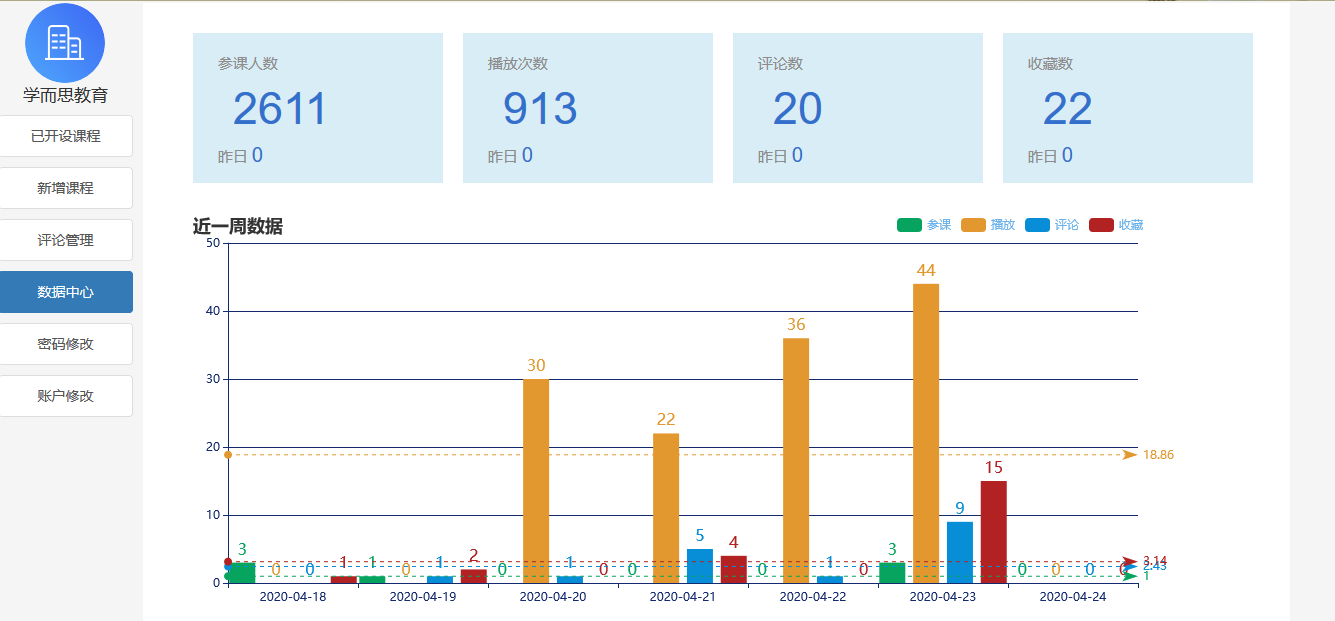


图6-20 机构数据中心界面

后台获取展示数据代码如下：

allJoinCourseNum = iInstitutionDataService.countAllJoinCourseNum (institutionId);

yesterdayJoinNum = iInstitutionDataService.countYesterdayJoinNum (institutionId, yesterday);

allPlayVideoNum = iInstitutionDataService.countAllPlayVideoNum (institutionId);

yesterdayPlayNum = iInstitutionDataService.countYesterdayPlayNum (institutionId, yesterday);

allCommentNum= iInstitutionDataService.countAllCommentNum (institutionId);

yesterdayComNum = iInstitutionDataService.countYesterdayComNum (institutionId, yesterday);

allCollectNum = iInstitutionDataService.countAllCollectNum(institutionId);

yesterdayColNum = iInstitutionDataService.countYesterdayColNum(institutionId, yesterday);

for(int i=1;i<=7;i++){

joinCourseList.add(iInstitutionDataService.countYesterdayJoinNum (institutionId, day));

playVideoList.add(iInstitutionDataService.countYesterdayPlayNum(institutionId, day));

commentList.add(iInstitutionDataService.countYesterdayComNum (institutionId,day));

collectList.add( iInstitutionDataService.countYesterdayColNum (institutionId, day));

}

6.14 首页轮播管理功能实现

管理员在轮播图管理界面，可以查看所有已经上传的轮播图信息，点击后面“展示”或“隐藏”按钮即可设置首页展示的轮播图片（如图6-21所示）。管理员还可以在此页面上传新的轮播图，新上传的轮播图默认状态为“展示”(如图6-22所示)。



图6-21 轮播图管理界面



图6-22 上传轮播图

分页获取轮播图的后台代码如下：

public ModelAndView carouselMag( int page, int size){

ModelAndView mv=new ModelAndView();

List<AdminUseCourse> allCourse= iAdminCarouselService.findAllCourse ();

List<AdminCarouselInfo> carousel = iAdminCarouselService.carousel(page, size);

PageInfo carouselListPageInfo=new PageInfo(carousel);

mv.addObject("allCourseNameId",allCourseNameId);

mv.addObject("carouselListPageInfo",carouselListPageInfo);

mv.setViewName("adminCarouselMag");

return mv;

}

上传轮播图代码和上传用户账户信息代码十分相似，故在此不再展示。

6.15 用户管理功能实现

管理员分别对机构账户和普通用户账户进行管理，在本模块中可以查看所有机构账户的状态，但只能对机构账户进行解封和停用的操作，其余操作需要移步至事务审核模块（如图6-23所示）。此外，鼠标悬停在资质照片上会自动放大显示，便于管理员审核机构资质。普通用户账户管理仅有解封和停用的操作，故在此不再详细阐述。



图6-23机构账户管理界面

放大查看资质图片功能代码如下：

$("img.dimg").mouseover(function () {

$(this).css("z-index","200");

$(this).css("transform","scale(5)");

});

$("img.dimg").mouseout(function () {

$(this).css("z-index","10");

$(this).css("transition","all .3s ease 0s");

$(this).css("transform","scale(1)");

});

解封功能后台代码如下：

@RequestMapping(value = "/institutionOpen")

public @ResponseBody

String institutionOpen(String institutionId){

try {

iAdminAccountMagService.openInstitutionAccount(institutionId);

return "1";

} catch (Exception e) {

return "0";

}

}

6.16 事务审批功能实现

事务审批分为机构注册审批和机构信息变更审批，本次以机构注册审批为例（如图6-24示），管理员点击通过后，该机构账户就可以正常登录，若点击拒绝按钮，则机构账户的状态变为“已终结，待修改重申”，等到机构账户再次登录时，系统会提醒机构账户修改注册信息，重新提交管理员审核。



图6-24 机构注册审批界面

本界面上通过和拒绝按钮的后端代码和机构账户管理的解封停用按钮的后端代码类似，只改变了账户的状态位，故在此不再粘贴代码。

6.17反馈处理功能实现

管理员未回复的用户反馈都会在未处理的用户反馈界面里面显示出来，如果反馈里所填的用户名和密码正确，则反馈状态为“验证通过”，反正则为“验证失败”（如图6-25示）。对于无意义的用户的反馈，管理可以选择忽略；如需回复反馈，点击回复按钮，跳出一个用户Email地址自动填充完成的回复框（如图6-26示），在此框内输入管理员回复的内容，点击发送即可。



图6-25 反馈处理界面

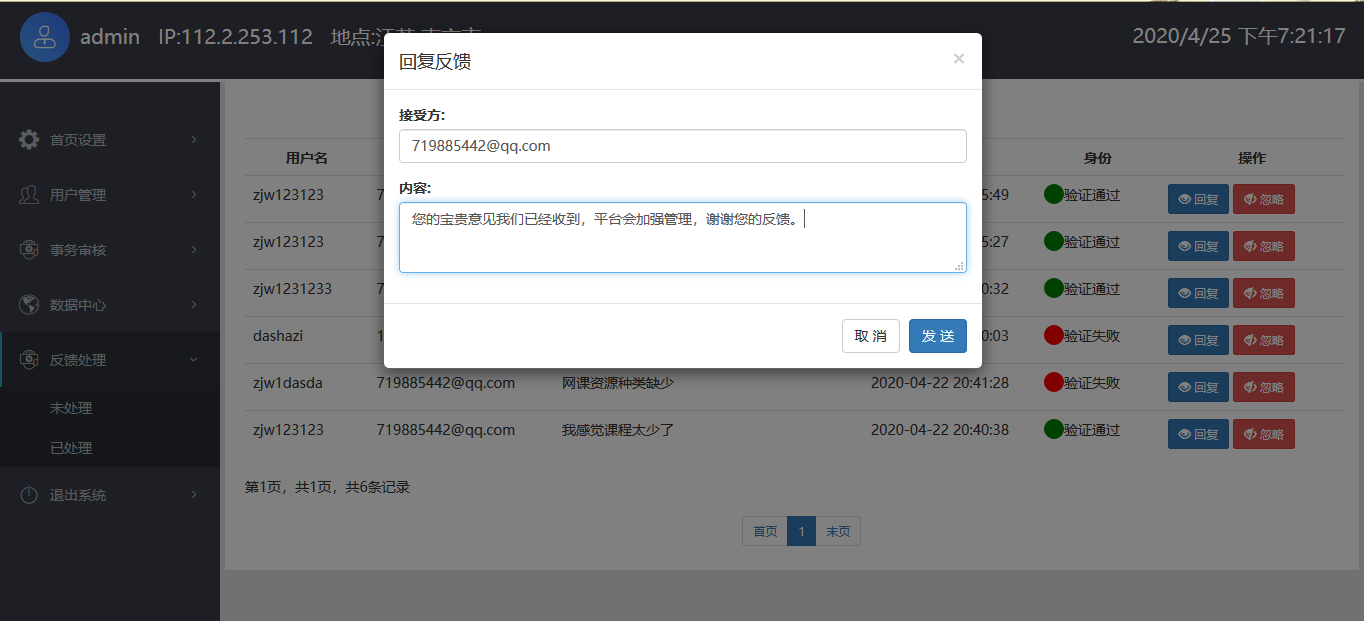


图6-26 反馈回复界面

管理员回复用户反馈后台代码如下：

@RequestMapping(value = "/handle")

public @ResponseBody String handle(String feedBackId,String reFeedBack,String email){

try {

iFeedBackService.handleFeedBack(feedBackId, reFeedBack);

SendEmail.sendMail(email,reFeedBack,"来自管理员的反馈回复");

return "1";

} catch (Exception e) {return "0"; }

}

6.18 后台数据中心功能实现

后台首页展示系统用户、机构、课程和评论总数，下方折现图展示当天各个时段登录量和播放量，旁边饼状图展示所有评论分数的分布（如图6-27所示）。全国数据展示使用地图展示各个省份登录的总人次，不同数量段在地图上显示出不同的颜色，便于区分查看（如图6-28所示）。昨日数据页面展示系统昨日允许情况，尤其是视频播放排名，有利于管理员掌握系统整体的倾向（如图6-29所示）。标签倾向界面展示用户和网课标签出现频率最高的10个词（如图6-30所示），管理员可以据此建议机构创建相关课程。

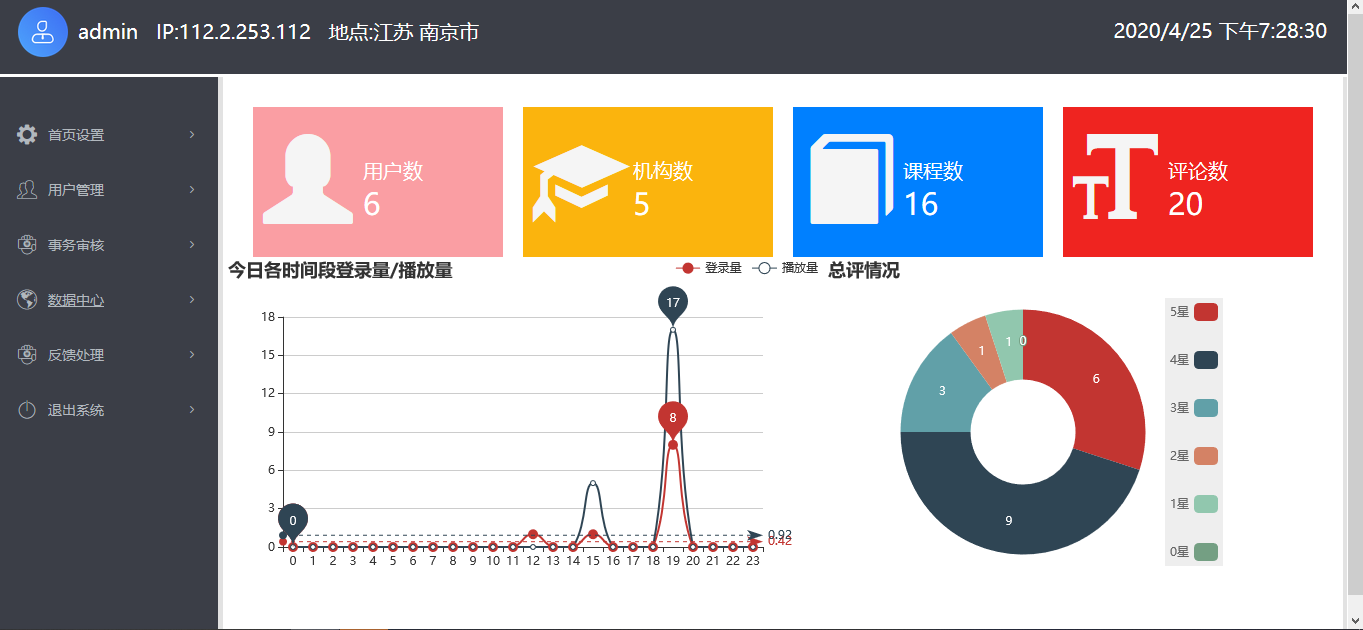


图6-27后台首页数据展示界面



图6-28 全国登录数据展示界面

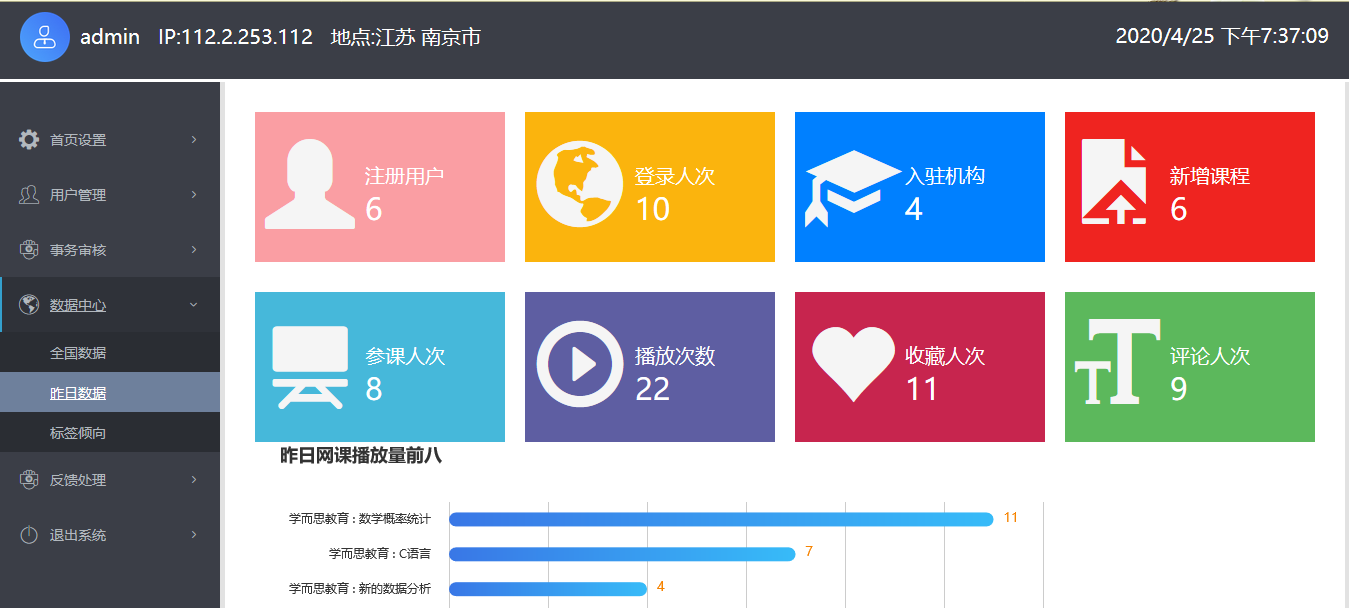


图6-29 昨日系统数据界面

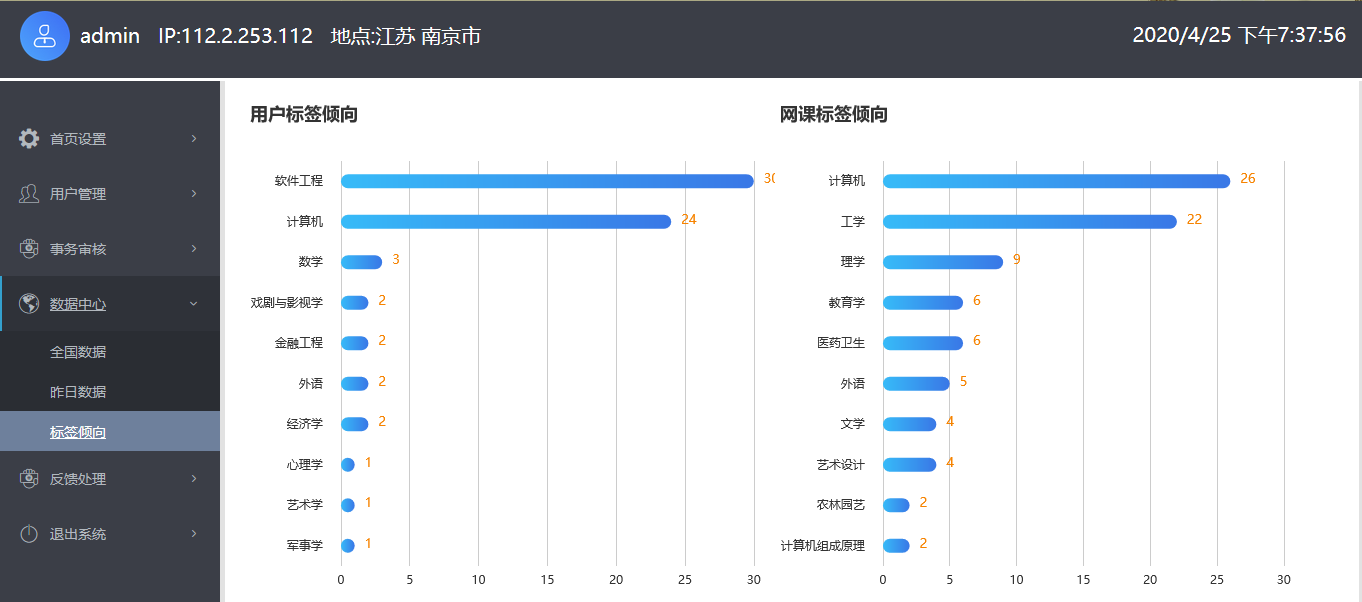


图6-30 用户和课程标签倾向对比界面

标签倾向后台代码如下：

List<String> allUserInterest = iAdminCountDataService.findAllUserInterest("");

Map<String, Integer> userInterestMap = Sort.getInterestAndNum(allUserInterest);

Map<String, Integer> sortUserInterestMapByValue = Sort.sortMapByValue(userInterestMap);

其中，allUserInterest为所有用户兴趣列表，userInterestMap的key为兴趣标签，value为此标签出现的次数，调用sortMapByValue()函数对用户兴趣map进行排序，取出前十存到结果map里，返回到前台展示。

统计标签出现次数后台代码如下：

public static Map<String,Integer> getInterestAndNum(List<String> allUserInterest){

List<String> interestList=new ArrayList<String>();

for(int i=0;i<allUserInterest.size();i++){

String[] userIntSplit = allUserInterest.get(i).split(",");

for(int j=0;j<userIntSplit.length;j++){

interestList.add(userIntSplit[j]); }

}

Set uniqueSet = new HashSet(interestList);

for (Object temp : uniqueSet) {

integerMap.put((String) temp,Collections.frequency(interestList, temp));

}

return integerMap;

}

6.19 系统部署

系统使用阿里云服务器进行部署，服务器IP地址为121.89.187.138，提供8080端口供外网访问，数据库版本为MySQL 5.5.62，Java语言开发工具包为JDK 13.0.1，Tomcat 8.5.50为Servlet容器。

在本地IntelliJ IDEA 2019.1.3内，将项目打包成war包，将war包复制到服务器Tomcat内的webapps文件夹下，再双击打开服务器Tomcat内bin文件夹下的startup.bat即可成功开启服务器，系统首页访问网址为<http://121.89.187.138:8080/onlineSchool/index.jsp，如图6-31>所示。

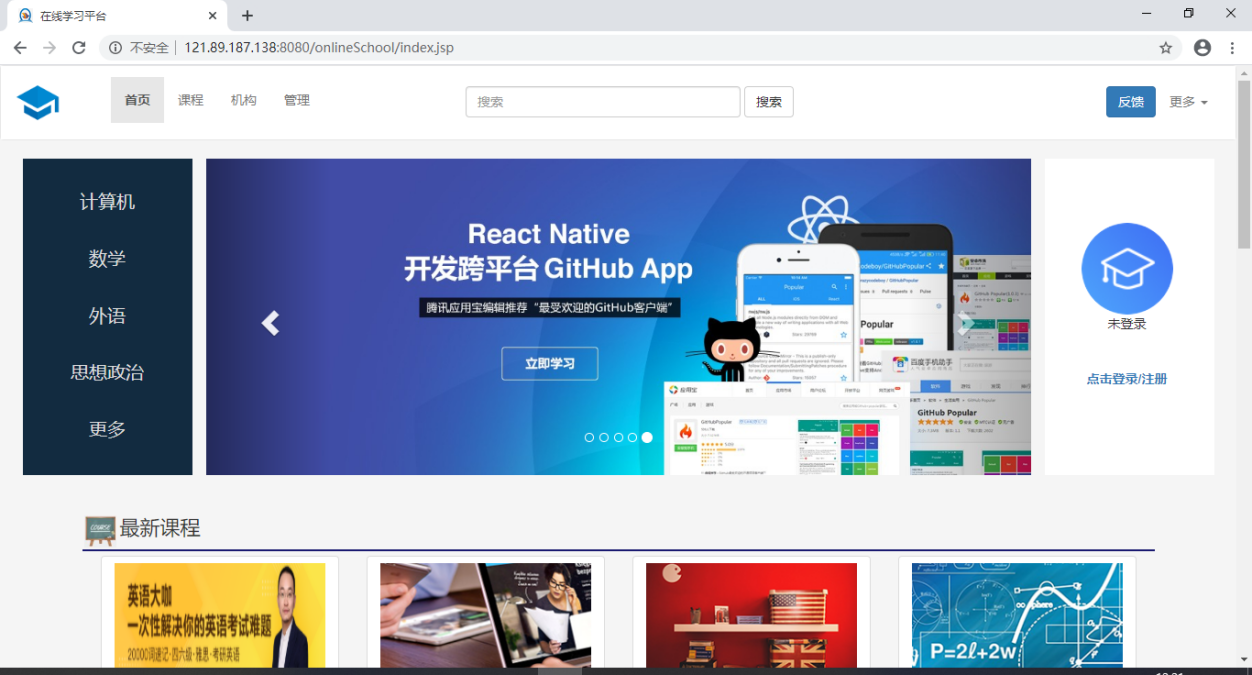


图6-31 系统首页展示

# 7 系统测试

系统测试采用黑盒测试方法，测试过程中检查每个功能是否可以正常使用。黑盒测试的重点在于测试程序的外部结构，并且首先检查软件界面和软件功能。

7.1 功能测试

对于系统几个重要的模块进行了功能测试，检测功能模块是否达到预期目标。首页课程推荐测试用例如表7-1所示。

表7-1 首页课程推荐模块测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 推荐模块测试 | | |
| 功能描述 | 验证能否根据用户填写的兴趣推荐课程 | | |
| 目的 | 向用户推荐可能感兴趣的课程 | | |
| 条件 | 用户正常访问网站首页 | | |
| 过程 | 查看未登录、登录和设置兴趣的情况下，主页课程推荐的内容 | | |
| 情况 | 测试用例 | 预期结果 | 实际情况 |
| 1 | 用户未登录 | 随机推荐八门网络课程 | 本次随机推荐为《计算机组成原理》、《C语言》、《数学概率统计》、《英语语音》、《系统局限性》、《Linux操作系统》、《数据分析》和《计算机硬件》，和预期结果一致 |
| 2 | 用户test001登录，但兴趣为空 | 随机推荐八门网络课程 | 本次随机推荐为《软件测试》、《单次速记》、《概率统计》、《英语语音》、《系统局限性》、《操作系统》、《设计模式》和《计算机硬件》，和预期结果一致 |
| 3 | 用户test002登录，兴趣为计算机科学、软件工程 | 推荐关键词与计算机或软件工程相关的八门网课 | 本次随机推荐为《计算机组成原理》、《C语言》、《软件测试》、《操作系统》、《Python》、《面向对象程序设计》、《系统设计与分析》和《计算机硬件》，和预期结果一致 |

视频上传测试用例如表7-2所示。

表7-2 视频上传测试用例表

|  |  |
| --- | --- |
| 名称 | 视频上传测试 |
| 功能描述 | 上传视频大于设置的上线，系统能否终止上传 |
| 目的 | 保证机构用户不能恶意上传大型视频文件 |
| 条件 | 机构账户通过审核，有权限上传视频 |
| 过程 | 上传不同大小的视频文件 |

续表7-2

|  |  |  |  |
| --- | --- | --- | --- |
| 情况 | 过程 | 预期结果 | 实际情况 |
| 1 | 上传300M以内的视频 | 点击上传，上传成功 | 和预期结果一致 |
| 2 | 上传1G的视频文件 | 点击上传，上传失败。 | 和预期结果一致 |

后台昨日数据展示界面测试用例如表7-3所示。

表7-3 后台昨日数据展示测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据展示测试 | | |
| 功能描述 | 前日播放课程视频不足8个 | | |
| 目的 | 检验容量为8的图标缺少数据的结果 | | |
| 条件 | 数据完整的正常显示，缺少数据的显示为0 | | |
| 过程 | 访问昨日数据界面 | | |
| 情况 | 过程 | 预期结果 | 实际情况 |
| 1 | 预制八个以上昨日视频播放记录 | 正常显示昨日播放量前八名的课程 | 和预期结果一致 |
| 2 | 预制八个以内但不为零个昨日视频播放记录 | 显示有数据项的数据，缺少数据项显示为空 | 和预期结果一致 |
| 3 | 删除所有昨日播放记录 | 显示图表结构，无数据 | 和预期结果一致 |

其他模块测试与之前几个模块测试类似，不再重复贴出。

7.2 易用性测试

易用性测试如表7-4所示。

表7-4 易用性测试

|  |  |  |
| --- | --- | --- |
| 测试用例 | 测试过程描述 | 测试结果 |
| 页面风格是否一致 | 网站Logo、页面布局、导航栏设计、按钮风格与分类、分页组件、字体大小、风格是否一致 | 系统页面风格一致 |
| 易浏览性 | 功能分区明显，用户能很快的找到自己想要的功能 | 页面功能布局符合使用习惯，导航栏放置快捷按钮 |
| 人性化的输入、输出设计，输出结果应简单、明了；尽可能减少用户输入 | 系统界面显示饱满，但又不拥挤，符合人的审美 |
| 交互界面布局充实，颜色搭配合理，风格一致，符合人们习惯 | 整体风格一致，符合人们习惯 |
| 易操作性 | 网站使用简单，重要地方有提示信息，异常操作能及时拦截 | 系统操作简便 |

# 8 总结

通过系统的设计与实现过程进一步掌握了在Java中使用SSM架构开发Web网页的知识，熟悉用Maven去导入架包、构建项目的过程，对本科阶段学习的软件开发的基本流程有了更深层次的理解，也对软件工程这一专业有更深的认识。

系统的开发首先要对用户的需求有准确的认识，然后通过软件工程开发语言将本系统的需求形成书面化、规范化的文档，及需求规格说明书；其次是对系统进行概要设计和详细设计，简单来说就是将整个系统拆分成若干个小的模块，并设计好程序的流程、使用的算法和数据库表的结构；然后进入系统编码阶段，选择Java的集成开发工具IDEA，规范了编码风格，提高开发效率。经过反复的测试，本系统满足了功能需求，完成预定的开发任务。但在开发过程中也暴露出一些自身能力的不足，尤其是数据库表设计上，由于设计阶段考虑不周，导致后面开发过程中，反复修改数据库表，大大延缓开发进度，有待进一步完善。本次毕业设计，是对本科阶段所学知识和动手能力的综合考察，一定程度上的增强了自身能力，收获颇丰。

# 参考文献：

[1] 任沁沁. 公安部发布最新全国机动车和驾驶人数据. 新华社，2021.12

[2] 刘志强. 停车场建设需再发力. 人民日报，2020.7

[3] Mujeeb Ur Rehman. A smart parking system to minimize search time, fuel consumption and CO2 emission. ICAC，2020.9

[4] MDN Contributors. MDN Web Doc Geolocation API. MDN，2022.4

[2] 刘宇宙. Python 3.5 从零开始学. 清华大学出版社，2017.8

[3] Miguel Grinberg. Flask Web Development. OReilly Media，2014.6

[4] 张海藩等. 软件工程导论. 清华大学出版社，2017.7

[5] 梁灏. Vue.js 实战. 清华大学出版社，2017.10

[6] Scott Chacon, Ben Straub. Pro Git. Apress，2021.8

[7] Antoni Roig Alegre. Scope of Parking in Europe. EPA, 2013.9

[8] James Turnbull. The Docker Book. Turnbull Press, 2014.7

# 致谢