****

**毕 业 设 计**

**题 目 基于Springboot的大型演唱会票务续订与vip服务平台的设计与实现**

**姓 名**

**学 号**

**学 院**

**专业年级**

**指导教师**

**2024 年 5月 30日**

中南林业科技大学涉外学院本科毕业设计诚信声明

本人声明：本人所交的中南林业科技大学涉外学院毕业设计，是在指导老师的指导下，独立完成的。该毕业设计不存在知识产权争议，除设计中已经注明引用的内容外，本设计不含任何其他个人或集体已经发表 或撰写过的作品成果。对本设计的研究做出重要贡献的个人和集体均已在文中明确标明。本人完全意识到本声明的法律结果由本人承担。

本科毕业设计作者签名：

2024年 5月 30日

**摘 要**

随着互联网技术的迅猛发展，电子商务已成为现代商业的重要组成部分。演唱会销售作为一种具有浓厚地域特色和文化内涵的商业活动，正逐渐从传统的实体店销售转向线上平台。基于Web的演唱会销售平台不仅可以帮助地方特色产品拓宽销售渠道，还能促进地方经济的发展，提高农民收入，推动乡村振兴。此外，演唱会销售平台还能有效地传播和弘扬地方文化，增强地域认同感，促进文化交流。这类平台不仅能够突破传统销售模式的地域限制，还能通过文化赋能提升产品附加值，但在实际运营中普遍存在演唱会分类混乱、文化属性弱化、用户体验割裂等问题。

本系统基于SpringBoot+Vue.js技术架构，构建了集演唱会展示、智能推荐于一体的演唱会销售平台。前端采用Vue.js实现响应式交互界面，后端通过SpringBoot微服务架构构建RESTful API，结合MySQL 8.0数据库集群与Redis缓存机制，形成高可用性解决方案。

**关键词**  演唱会销售平台；SpringBoot框架；Vue框架；MySQL数据库

**Title** Design and Implementation of a Specialty Product Sales Platform

**Abstract:**

With the rapid development of internet technology, e-commerce has become an integral part of modern business. Specialty product sales, as a commercial activity with strong regional characteristics and cultural connotations, are gradually transitioning from traditional brick-and-mortar stores to online platforms. A web-based specialty product sales platform not only helps expand the sales channels for local products but also promotes regional economic development, increases farmers' income, and drives rural revitalization. Furthermore, such platforms can effectively disseminate and promote local culture, enhance regional identity, and foster cultural exchange. These platforms can break through the geographical limitations of traditional sales models and increase product value through cultural empowerment. However, common issues in their operation include chaotic product categorization, weakened cultural attributes, and fragmented user experiences.

This system is built on the SpringBoot + Vue.js technology stack, creating a specialty product sales platform that integrates product display and intelligent recommendation. The front end utilizes Vue.js to implement a responsive interactive interface, while the back end employs the SpringBoot microservices architecture to construct RESTful APIs. Combined with a MySQL 8.0 database cluster and Redis caching mechanism, the system forms a highly available solution.

**Keywords:** Specialty product sales platform; SpringBoot framework; Vue framework; MySQL database

目 录

[1 绪论 1](#_Toc18486)

[1.1 课题研究背景 1](#_Toc12874)

[1.2 课题研究现状 1](#_Toc20598)

[1.3 本文主要工作 2](#_Toc26279)

[1.4 本文组织结构 3](#_Toc35)

[2 技术简介 4](#_Toc21749)

[2.1 B/S架构 4](#_Toc4706)

[2.2 前端框架Vue 4](#_Toc1889)

[2.3 后端框架Spring Boot 4](#_Toc15809)

[2.4 数据库MySQL 6](#_Toc17139)

[2.5 开发工具IDEA 6](#_Toc16665)

[3 演唱会票务续订与vip服务平台分析与设计 7](#_Toc18792)

[3.1 需求分析 7](#_Toc25704)

[3.2 系统可行性分析 1](#_Toc786)3

[3.3 系统数据流图 15](#_Toc19081)

[3.4 数据库设计 15](#_Toc971)

[4 演唱会票务续订与vip服务平台实现 27](#_Toc20353)

[4.1 系统目录结构 27](#_Toc6107)

[4.2 系统前端功能模块的实现 27](#_Toc18055)

[4.3 系统后端功能模块的实现 40](#_Toc17608)

[5 系统测试 50](#_Toc1772)

[5.1 测试的原理 50](#_Toc14011)

[5.2 系统测试用例 50](#_Toc32200)

[5.3 测试结果分析 56](#_Toc14755)

[5.4 本章小结 56](#_Toc21951)

[6 总结与展望 57](#_Toc19028)

[6.1 总结 57](#_Toc10763)

[6.2 展望 57](#_Toc18929)

[致 谢 59](#_Toc1744)

[参考文献 60](#_Toc11634)

# 

# **1** 绪论

## **1.1** 课题研究背景

随着互联网技术的迅猛发展，数字化服务已经渗透到人们日常生活的方方面面。在娱乐产业中，演唱会作为大众文化的重要组成部分，其消费模式也在经历着深刻的变革。传统的演唱会票购买方式往往需要观众亲自前往演唱会院售票窗口排队购票，这种方式不仅耗时费力，而且难以满足现代消费者对于便捷性与个性化的需求。与此同时，网络平台上的演唱会评论成为了观众选择影片的重要参考依据之一，它能够帮助潜在观影者快速了解演唱会内容及质量，从而做出更加合理的观影决策。

近年来，在线购票服务迅速兴起，并逐渐成为主流趋势。各大演唱会院纷纷推出了自己的在线选座系统或是与第三方平台合作提供网上订票功能，极大地提升了用户体验和服务效率。然而，市场上现有的很多解决方案存在着诸如界面设计不够友好、支付流程繁琐、安全性不足等问题，这些问题限制了进一步提升用户满意度的可能性。此外，尽管许多网站或应用程序提供了评论区供用户发表意见，但如何有效管理和利用这些海量信息，以更好地服务于广大影迷群体，则是当前面临的一大挑战。

基于此背景下，本毕业设计旨在构建一个基于Spring Boot框架开发的演唱会售票与评论管理系统。该系统将结合云计算、大数据等前沿信息技术，致力于打造一个集高效便捷购票体验与丰富互动交流于一体的综合服务平台。通过优化前端页面布局和交互逻辑，简化购物流程，确保交易安全可靠；同时引入智能推荐算法对用户行为进行分析挖掘，实现精准化营销推广。更重要的是，我们将建立一套完善的评论管理体系，包括但不限于敏感词过滤、垃圾信息识别等功能，保证社区环境健康积极向上，促进良好文化氛围形成。总之，该项目不仅具有重要的实际应用价值，也为推动我国文化产业创新发展贡献了一份力量。

## **1.2** 课题研究现状

自随着互联网技术的迅猛发展，演唱会行业的数字化转型已成为全球趋势。在欧美等发达国家，在线演唱会售票系统和评论平台已相当成熟，并不断通过技术创新提升用户体验和服务质量。例如，Ticketmaster和Fandango不仅提供演唱会票预订服务，还利用大数据分析实现个性化推荐；IMDb、Rotten Tomatoes和Metacritic等平台为观众提供了全面客观的演唱会评价体系；部分演唱会院尝试引入VR头显设备，探索虚拟现实体验的应用。

在中国，演唱会产业也经历了深刻的变革，猫眼演唱会、淘票票等在线票务平台成为连接观众与影院的重要桥梁。这些平台利用大数据分析为用户提供个性化的影片推荐，整合丰富的演唱会资讯，构建了全面的服务生态系统。此外，移动支付集成、人工智能应用以及云计算支持等技术手段的应用，使得购票过程更加便捷安全，提升了服务质量。豆瓣演唱会作为最受欢迎的演唱会评价社区之一，用户可以在此发表影评并根据个人喜好给影片打分，综合考虑专业人士意见和大众反馈生成总体评分，为潜在观影者提供了重要参考。

总的来说，基于现代Java框架如Spring Boot开发的演唱会票务续订与vip服务平台在全球范围内都取得了显著的发展。无论是国外的Ticketmaster、Fandango还是国内的猫眼演唱会、淘票票，都在努力通过技术创新和服务优化来满足用户需求。国外平台更侧重于娱乐活动的多样性及社交功能的拓展，而国内平台则充分利用本土化优势，如移动支付、社交媒体集成等，提供一站式的购票解决方案。

演唱会评论方面，国外的IMDb、Rotten Tomatoes以其独特的评分机制影响着全球观众的选择，而中国的豆瓣演唱会则以详尽的资料库和用户生成内容吸引了大量忠实用户。微博演唱会凭借其强大的社交属性成为了传播热门话题和明星动态的重要渠道，促进了信息共享和文化讨论。

未来的研究和发展需关注如何进一步提升服务质量、增强安全性以及探索新的商业模式。特别是在中国这样快速发展的市场中，结合本地特色进行创新显得尤为重要。持续跟踪国际前沿动态，借鉴成功案例的同时注重差异化发展将是推动这一领域不断前进的关键所在。随着技术的进步和市场的变化，演唱会票务续订与vip服务平台将继续演进，为用户提供更加丰富多样的服务体验。

## 1.3 本文主要工作

总项目可分为两部分进行；

首先，进行系统的开发实现工作。本系统的构建采用B/S架构，并运用JAVA语言的Springboot框架以及IDEA开发工具来进行系统的开发实现工作。同时，为了支持项目的顺利进行都选择当前流行的技术框架。在数据存储的设计中选定MySQL数据库作为本次小型项目的核心数据存储方案。

然后，进行论文文档编写工作，需要对课题背景进行深入的研究，并进行社会调查，以全面理解项目的实际需求。再对系统进行详细的设计与分析，并展示系统的各项功能。在完成系统功能的展示后对系统进行全面的测试，以确保其稳定性和可靠性。最后，对整个项目进行总结，提炼出有价值的经验和教训。

## 1.4 本文组织结构

本论文首先在第一章绪论中，探讨了研究演唱会票务续订与vip服务平台课题的背景和当前现状，为研究的可行性奠定基础。同时，概述了论文的整体结构和相关工作。

第二章技术简介部分，详细介绍了开发演唱会票务续订与vip服务平台所需的技术，包括B/S模式、前端vue框架和后端spring boot技术等。

第三章系统分析与设计，首先对项目进行了系统的总体需求分析，接着对可行性进行了详细分析，为后续实现打下基础。同时，本章还涵盖了数据库的设计，包括概念结构设计、逻辑结构设计、以及数据库表的设计。

第四章系统实现，着重介绍了前端界面和后端代码的实现过程。

第五章系统测试，对系统进行了功能模块测试、和整体系统测试，最后进行验收测试，确保系统满足用户需求。

最后，第六章总结与展望，为整个论文的研究工作画上了圆满的句号。

# **2** 相关技术简介及部署环境说明

## 2.1 Java语言

Java是一种面向对象的静态式编程语言。Java编程语言具有多线程和对象定向的特点。其特点是根据方案的属性将方案分为几个不同的模块，这些模块是封闭的和多样化的，在申请过程中具有很强的独立性。Java语言在计算机软件开发过程中的运用可以达到交互操作的目的，通过各种形式的交换，可以有效地处理所需的数据，从而确保计算机软件开发的可控性和可见性。开发Java语言时，保留了网络接口，Java保留的缺省网络接口可以与web应用程序编程所依赖的类别库相匹配。为了使Java开发的应用程序更加稳定和强健，Java会自动收集程序中的垃圾，并处理程序中存在的异常。Java语言是日常开发过程中广泛使用的通用基本语言。其中Java语言演唱会库、句子、语法规则和关键字经常用于计算机软件的开发和编程。

面向对象编程是Java语言最显着的特点。它具有原始接口和补充接口以及继承，不仅可以实现相同类型的单个继承，而且还支持接口之间的多个继承，从而实现类、接口和接口之间以及类和接口之间的有效通信。Java的面向对象特性主要包括三个方面:继承、多态性和封装。封装是Java的核心，可以封装所有数据操作。多态性是指由面向对象行为派生的相关行为。继承作为特殊编程模式有两种类型:父类和子类，这两种类型的属性具有相同的功能和特性。对于父类的属性特性，子类可以实现继承和优化。

## 2.2 MySQL简介

MySQL是一种关系型的数据库管理系统，属于Oracle旗下的产品。MySQL的语言是非结构化的，使用的用户可以在数据上进行工作。这个数据库管理系统一经问世就受到了社会的广泛关注。在各个方面，与同等的数据库相比，MySQL的优点极为突出，它的运行速度快，适用的范围广泛，而且数据库的安全性这一方面独树一帜。在语言结构方面，MySQL的语言简单，其他数据库需要一大段代码来实现的操作，MySQL仅需要一小部分代码甚至几行。综上所述，MySQL这种关系型数据库管理系统，已经成为了开发者进行项目的数据开发、存储的不二之选。MySQL的功能也多种多样，如数据操纵和数据库的建立维护等。而且该数据库的数据共享性高、冗余度低而且容易扩充。MySQL在安全性这一方面也具有自身的特点，它应用了用户的标识和鉴别技术，对试图和数据进行加密，确保资料信息的可靠性。介于数据库系统的功能与强大等性质之间，本数据库系统的设计中主要使用了MySQL实现对数据的处理。本系统运用MySQL数据库，在Web应用这一块，MySQL是最好的选择。对于该系统整个的开发、搭建、运行和维护具有极其重要的作用。

## 2.3 后端框架Spring Boot

Spring Boot是由Pivota公司的团队在Spring框架的基础上所开发出的一个轻量级应用框架，其设计目的是为了简化使用传统Spring框架开发时初始搭建、配置、开发、测试和部署等复杂的流程，满足快速、敏捷开发的需要[[49]。传统Spring框架开启某些特性或引入第三方依赖时，需要配置大量的XML文件，比较繁琐且非常容易出错。而SpringBoot的理念则是“约定大于配置”，它集成了大量第三方依赖的配置，再配合内部自动装配的机制，仅需在Maven工程的POM文件中添加相关依赖即可实现开箱即用的效果，这使得开发人员更加专注于实现业务逻辑，而不是繁琐的配置。此外，Spring Boot内置了Servlet容器如Tomcat和Jetty，项目可以直接通过命令在本地启动，无需打包成文件再部署到服务器上，大大减少了测试和部署的成本。目前，Spring Boot己经成为了后端服务器应用开发的主流技术，其特点和优势总结如下:

(1)简化配置，遵循“约定大于配置”的理念，减少了样板代码、注释和XML配置文件的编写。

(2)内置Servlet容器，可在本地独立运行项目，方便测试和部署。

(3)自动装配，可以非常方便地与Spring生态系统中其他框架或第三方依赖集成。

(4)便于快速构建项目，尤其适合构建本系统。

## 2.4 数据库MySQL

MySQL是一种关系型的数据库管理系统，属于Oracle旗下的产品。MySQL的语言是非结构化的，使用的用户可以在数据上进行工作。这个数据库管理系统一经问世就受到了社会的广泛关注。在各个方面，与同等的数据库相比，MySQL的优点极为突出，它的运行速度快，适用的范围广泛，而且数据库的安全性这一方面独树一帜。在语言结构方面，MySQL的语言简单，其他数据库需要一大段代码来实现的操作，MySQL仅需要一小部分代码甚至几行。综上所述，MySQL这种关系型数据库管理系统，已经成为了开发者进行项目的数据开发、存储的不二之选。MySQL的功能也多种多样，如数据操纵和数据库的建立维护等。而且该数据库的数据共享性高、冗余度低而且容易扩充。MySQL在安全性这一方面也具有自身的特点，它应用了用户的标识和鉴别技术，对试图和数据进行加密，确保资料信息的可靠性。介于数据库系统的功能与强大等性质之间，本数据库系统的设计中主要使用了MySQL实现对数据的处理。本系统运用MySQL数据库，在Web应用这一块，MySQL是最好的选择。对于该系统整个的开发、搭建、运行和维护具有极其重要的作用。

## 2.5 容器平台Docker

Docker是一种开源的容器化平台，能够将应用程序及其依赖打包成一个轻量级、可移植的容器。与传统的虚拟化技术不同，Docker容器共享宿主机的操作系统内核，从而显著减少了资源占用和启动时间。Docker的核心优势在于其高效性、一致性和可移植性。通过Docker，开发者可以在本地开发环境中构建容器，并将其无缝部署到生产环境中，确保环境的一致性。此外，Docker还提供了强大的镜像管理、网络配置和存储管理功能，支持多容器协同工作。在演唱会票务与会员服务平台的开发中，Docker可以用于快速部署和测试，提高开发效率和系统稳定性。Docker还支持与Kubernetes等容器编排工具集成，能够实现大规模应用的自动化部署和管理。其轻量级特性使得Docker在云计算和微服务架构中得到了广泛应用，成为现代软件开发的重要工具。

## 2.6 系统的开发环境

系统使用Java语言，以SPRINGBOOT作为框架进行开发，其所使用的硬件和软件环境如下：

硬件环境：Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz 2.50 GHz 处理器，16.0GB 运行内存。

软件环境：Windows10 64位操作系统，IDEA后端开发工具，VScode前端开发工具。

# **3** 系统分析与设计

## 3.1 需求分析

需求分析在系统开发中有着十分重要的作用。软件项目凭借软件工程的思想和步骤可以大大的提高开发效率，缩短软件开发周期，保证了软件项目的质量。需求分析作为软件工程方法中的一步是至关重要的。软件需求工程是一门分析并记录软件需求的学科。需求分析简单的来说就是用户需要什么，系统需要什么，对此进行问题的列举，等级的排列，需要缜密的思分析和大量的调研。

演唱会票务续订与vip服务平台在国内有很多值得借鉴的例子，功能也都趋于完善，因此此次演唱会票务续订与vip服务平台将轻量化开发，要完成以下功能：

(1)票务续订系统的智能化设计：深入研究并设计一套高效、智能化的票务续订流程，确保用户在续订过程中能够享受到便捷、快速的操作体验。这包括用户身份验证的简化、票务信息的实时更新与查询、续订操作的优化以及支付环节的安全与便捷性提升。

(2)VIP服务平台的个性化开发：针对VIP客户，开发一个功能丰富、界面友好的服务平台。该平台将提供个性化的服务选项，如优先购票权、专属座位选择、定制化服务推荐等，以满足VIP客户的独特需求和期望。同时，还将设计一套完善的VIP会员管理系统，用于管理会员等级、积分累计与兑换、会员特权等。

(3)系统稳定性与安全性保障策略：深入研究并实施一系列有效的安全措施，如数据加密技术、访问控制机制、安全审计流程等，以确保用户数据的安全性和隐私保护。同时，还将对系统进行全面的性能测试，包括压力测试、并发测试等，以评估系统在高负载情况下的稳定性和可靠性。

(4)系统可扩展性与可维护性设计：在设计系统时，将充分考虑未来的扩展性和可维护性需求。通过采用模块化设计、灵活的系统架构以及完善的文档和代码规范，确保系统能够轻松应对未来功能的扩展和升级，同时降低维护成本和提高维护效率。系统功能模块如图3.1所示：

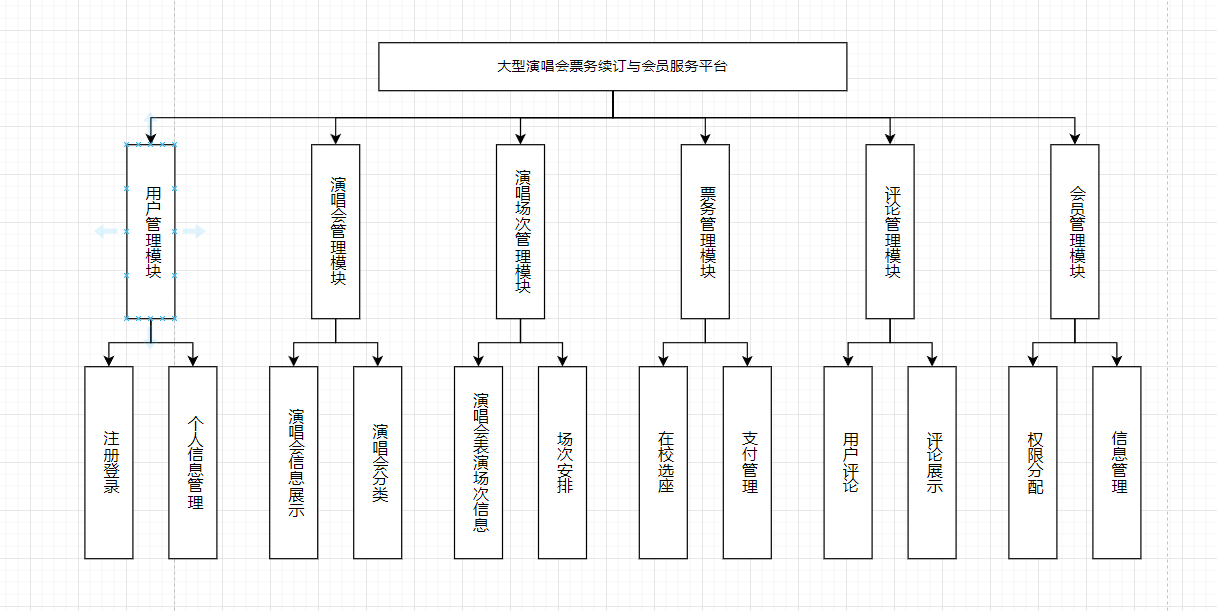


图3.1 系统功能模块图

以上是演唱会票务续订与vip服务平台的一些主要功能模块，这些模块相互协作，共同构成了整个系统的核心功能。通过这些功能，演唱会票务续订与vip服务平台能够为演唱会售卖提供高效的售卖方案，增加演唱会售卖的渠道。具体的功能说明如下；

系统管理员权限可以操作全部功能模块，影院管理员和评论管理员能审核演唱会信息和管理。用户管理员只能录入信息等。

1. 用户管理模块

模块功能：实现用户的登录、个人信息维护等功能。

本模块的主要功能是对个人信息进行增删改查操作。

(2) 演唱会管理模块

模块功能：展示演唱会信息、搜索和筛选演唱会。

这一模块的功能主要是对演唱会信息进行管理，如对项目的名称、简介、主演等信息进行管理，必要时进行增加、查询和删除

(3) 演出场次管理模块

模块功能：展示影院信息、场次安排。

这个模块的功能是对演唱会的添加、审核、删除等操作。

1. 票务服务模块

模块功能：在线选座、支付、订单管理。

这一模块的功能主要查看演唱会片的数量，还有支付结果和预约情况。

(5) 评论系统模块

模块功能：用户评论提交、评论审核、评论展示。

这一模块的功能主要是对用户评论进行添加，修改，查询操作。

(6) 推荐系统模块

模块功能：基于用户行为的个性化推荐。

这一模块的功能主要是对用户的个性化进行管理。

具体用例图如图3.2、图3.3所示：

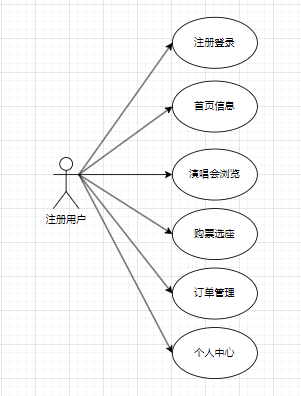


图3.2 用户用例图

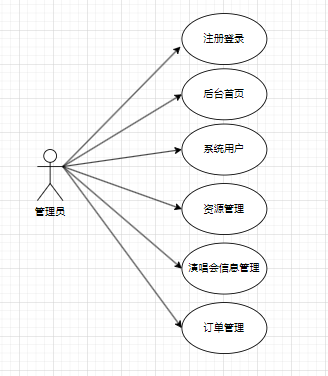


图3.3 管理员用例图

用例描述如表3.1、表3.2、表3.3所示：

表3.1 演唱会浏览操作用例描述表

|  |  |
| --- | --- |
| **用例名称**： | 演唱会浏览操作 |
| **参与者**： | 用户查看演唱会列表和分类 |
| **简要说明**： | 用户在购买前，对演唱会进行浏览操作 |
| **前置条件：** | 用户成功注册和登录平台 |
| 基本事件流： | 1. 用户选择分类/搜索关键词  2. 系统返回演唱会列表  3. 用户查看演唱会详情 |
| 后置条件： | 显示演唱会信息 |

表3.2演唱会票下单购买用例描述表

|  |  |
| --- | --- |
| 用例名称： | 演唱会票下单购买 |
| 参与者： | 用户 |
| 简要说明： | 用户下单购买演唱会票 |
| 前置条件： | 用户成功登录 |
| 基本事件流： | 1. 用户选择演唱会场次 2. 选座座位 3. 提交订单 |
| 后置条件： | 生成待支付订单 |

表3.3订单状态查询用例描述表

|  |  |
| --- | --- |
| 用例名称： | 订单状态查询 |
| 参与者： | 用户 |
| 简要说明： | 用户查看已经下单的状态 |
| 前置条件： | 用户有历史订单 |
| 基本事件流： | 1. 用户进入订单列表  2. 选择特定订单查看详情  3. 显示对应二维码信息 |
| 后置条件： | 显示订单详细信息 |

其中部分功能描述如下：

(1) ​个人信息管理：用户可在个人中心查看和修改昵称、头像、密码等基础信息，管理收货地址（增删改查及设置默认地址），并支持更换绑定手机号或邮箱。

(2) ​演唱会浏览与分类查询：用户可通过演唱会分类（如“流行”）导航筛选演唱会，并查看演唱会详情（含描述、价格等）。

(3) ​下单与支付：用户选择演唱会规格及数量后，系统实时校验并生成待支付订单，需完成支付（如微信/支付宝），支付成功则订单状态变更为“待发货”，超时未支付自动取消订单。

(4) ​订单状态追踪：用户可查看订单状态，待发货订单支持取消或退款，已购买可查看详情并进行演唱会评价。

(5) ​商家管理功能：商家可管理商品信息（增删改查、上下架及设置推荐状态），订单，审核用户退款申请（验证订单状态后回调支付系统退款）。

用户登录与场次查询时序图如图3.5所示：

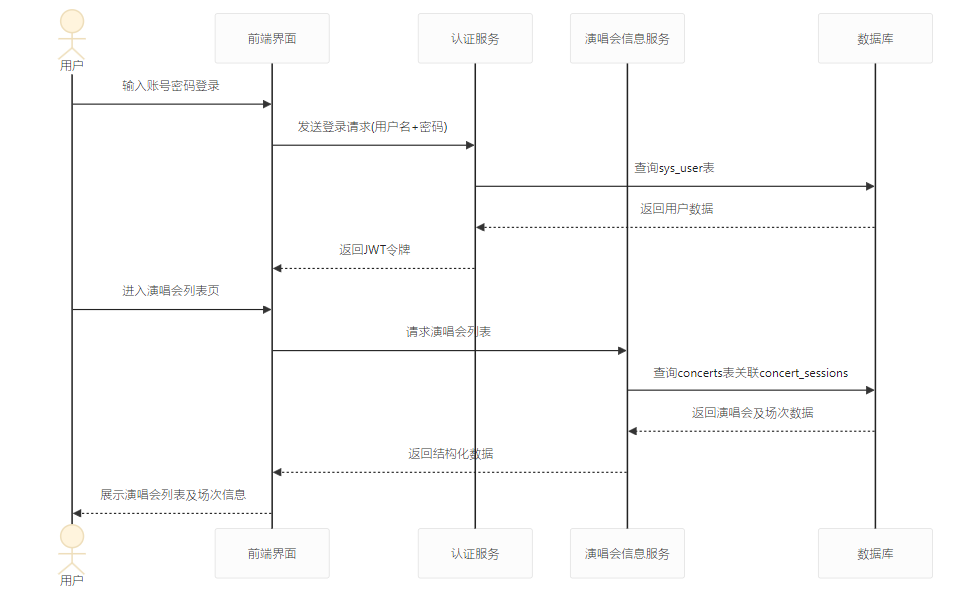
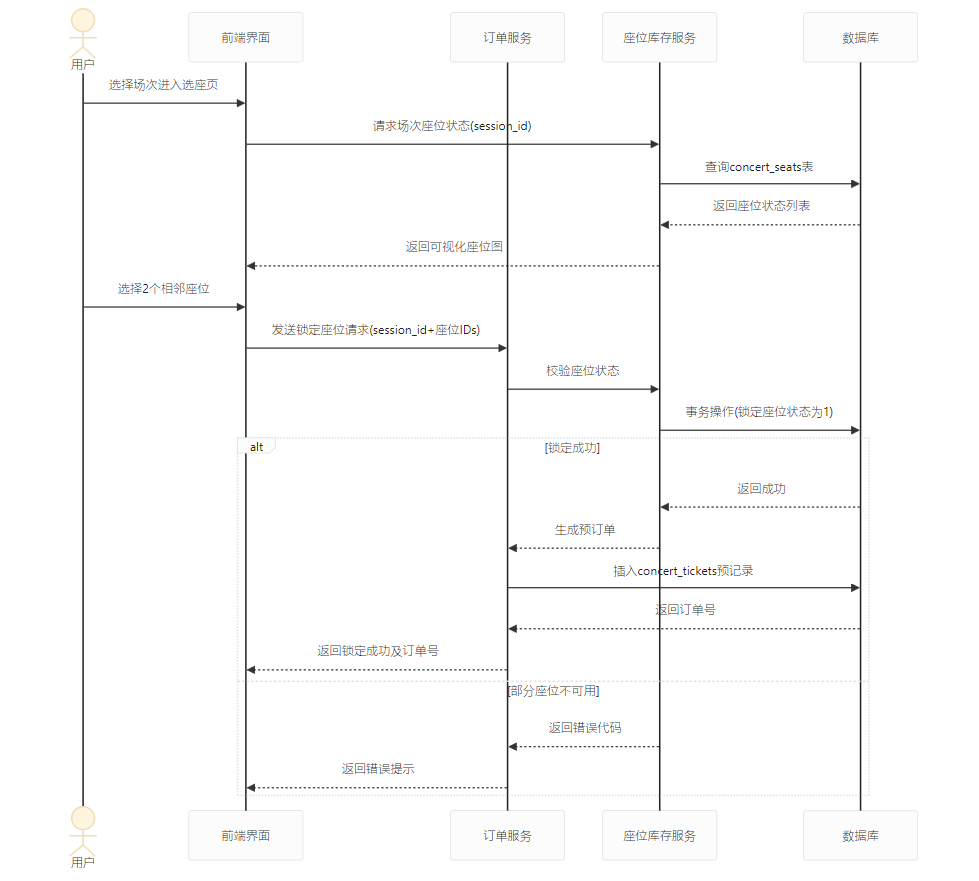
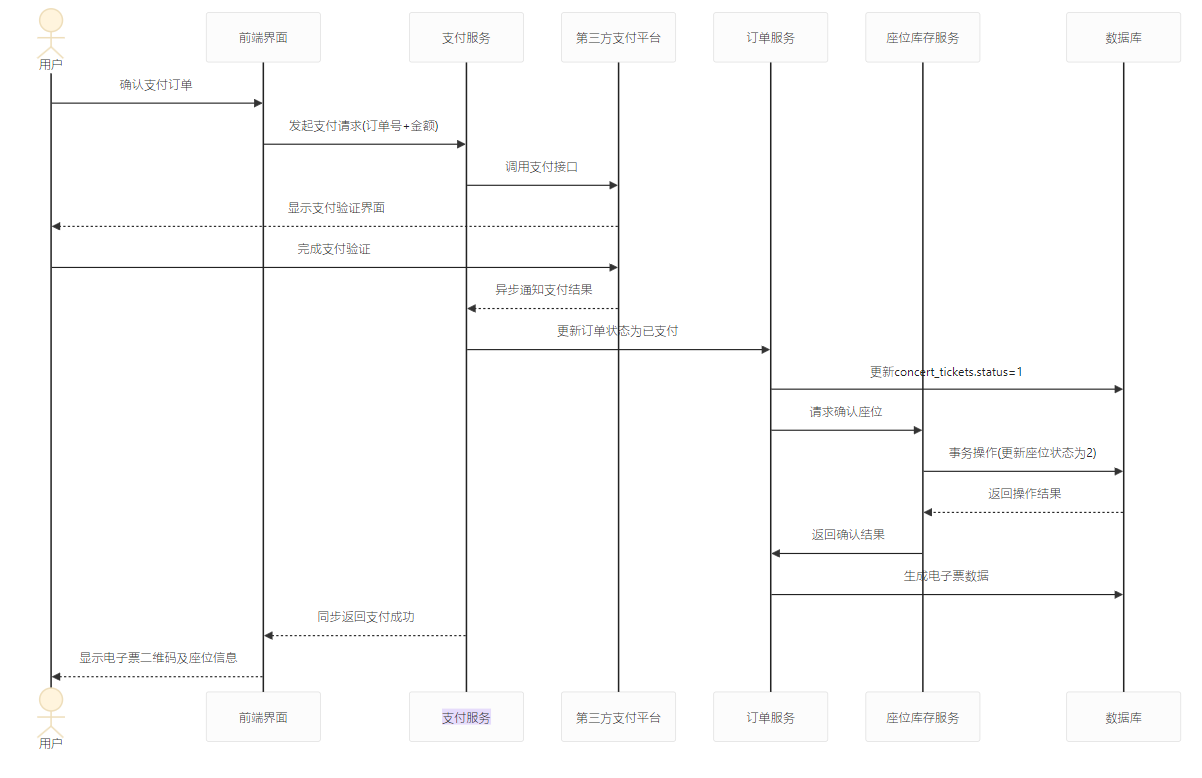


图3.4 用户登录与场次查询时序图

座位选择与锁定时序图如图3.5所示。

图3.5 座位选择与锁定时序图

订单支付与出票时序图如图3.6所示。

图3.6 订单支付与出票时序图

## 3.2 系统可行性分析

演唱会票务续订与vip服务平台的可行性分析基于当下的互联网背景，从经济、技术、法律和用户使用上进行了调查，从此验证次系统开发的可行性[6]。下面分别从以下几点进行分析:

（1）经济可行性分析：此系统所用的框架技术完全是开源的，其余的软件使用也都是免费的，在开发方面所消耗的成本可以忽略不计。而建立系统信息库所需的费用远远少于其带来的社会收益，从经济上分析系统是完全可行的。

（2）技术可行性分析：对系统进行分析，演唱会票务续订与vip服务平台的大部分功能都需要管理员完成，系统只需要添加少量代码就可以很好适配管理员功能[7]。可以提高文件的复用率，提高效率。在数据库方面，ORM简化了表的创建和使用，采用了Oracle旗下开源的关系型数据管理系统MySQL，多表存储的特点使数据更加规范化，增删改查更容易[8]。综上所述，技术开发上并无太大的难点。

（3）用户使用可行性分析：鉴于系统的使用用户，有统一的账号和密码，且系统无需用户有过多的操作，界面清晰简洁，用户使用可行性上无问题。

（4）法律可行性分析：即分析本系统是否与各类法律相悖。本系统使用市面开源免费软件开发，且作为个人毕设，无商用，均为本人自主开发，并且页面设计合理，发布的信息要求符合常规。整个系统无抵触法律法规的问题。因此在法律上，本系统可行。

## 3.3 系统数据流图

数据流图是阐述数据在软件内部流转与处理逻辑过程的图示，采用特定的图形元素来呈现这些数据流。它详尽地揭示了系统内各个处理环节间的数据关联，是结构化系统分析手法中不可或缺的重要工具之一。通过数据流图能够更直观地理解数据在系统中的流动与处理情况。本系统的数据流图如图3.8所示：

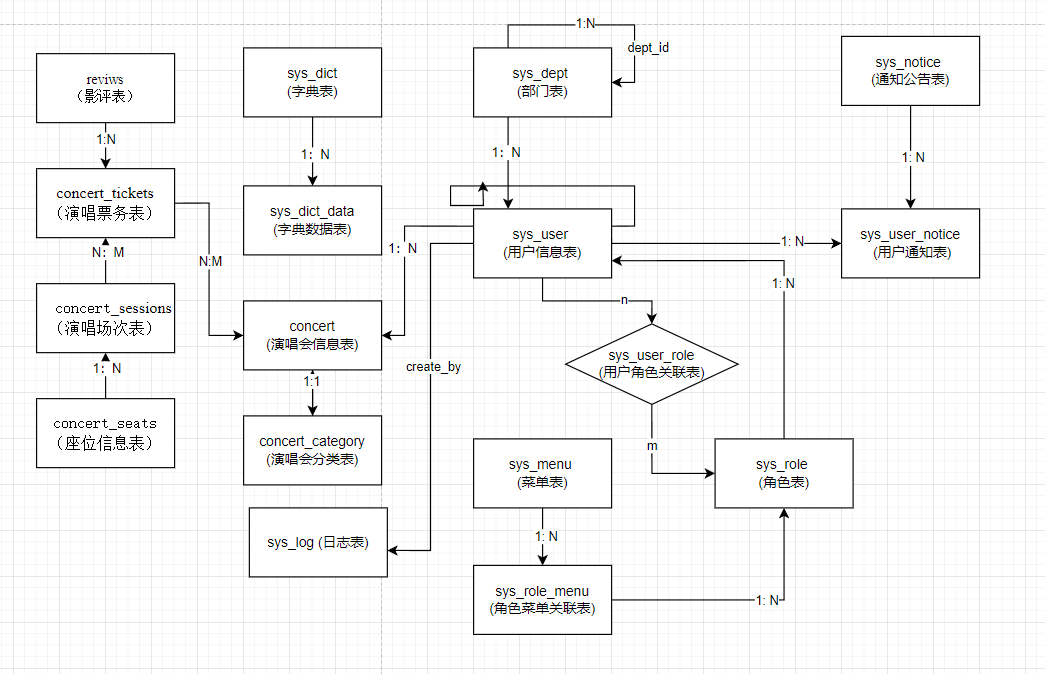


图3.8 系统数据流图

## 3.4 数据库设计

3.4.1 概念结构设计

在需求分析阶段，我们收集到了实现系统所需的基本数据和信息。为确保这些数据能更精确地反映系统特性，需要构建一个概念数据模型。此模型依据本系统的信息构建，能映射现实环境，进而促进对系统的深入理解和灵活调整。通常会采用E-R（实体-关系）模型来阐述这一概念模型，它由数据实体、实体的属性及实体间的关联关系共同构成。本系统的E-R图如图3.9所示：

图3.9 系统ER图

1. 部门实体（sys\_dept）​​

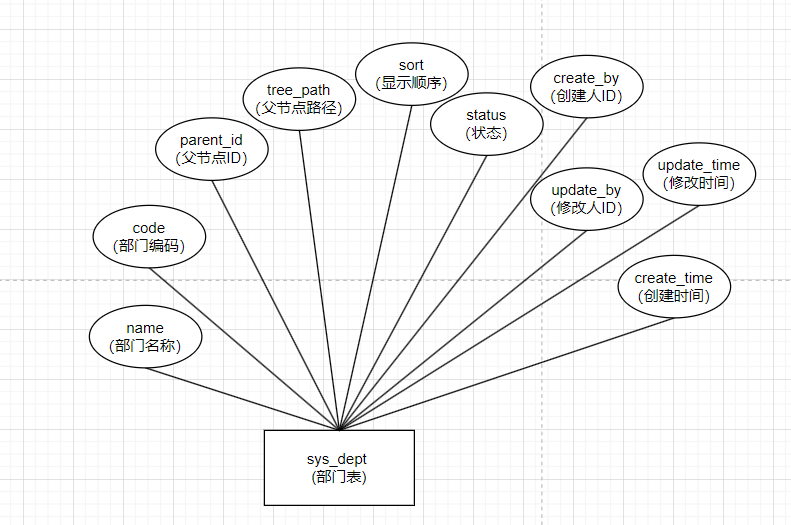
主要包含部门名称、部门编号、父节点ID、层级路径、显示顺序、状态等属性，用于实现组织架构的树形管理。  


图3.10 部门实体属性图

1. 字典实体（sys\_dict）​​

包含字典编码、类型名称、状态、备注等属性，支持系统动态配置字典类型，如性别、通知类型等。

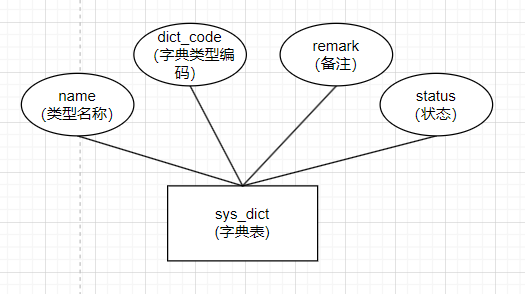


图3.11 字典实体属性图

1. 菜单实体（sys\_menu）​​

包含菜单名称、类型（目录/菜单/外链/按钮）、路由路径、组件路径、权限标识、图标等属性，用于构建动态路由和权限控制。

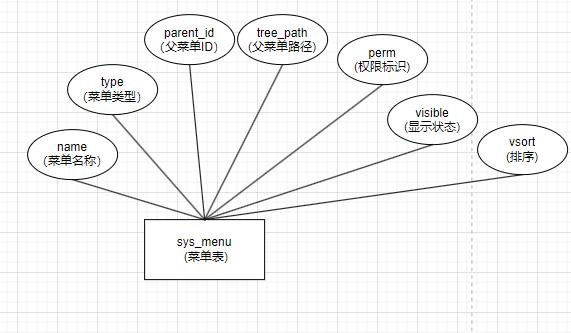


图3.12 菜单实体属性图

1. 角色实体（sys\_role）​​

包含角色名称、角色编码、数据权限范围、状态等属性，支持基于角色的访问控制（RBAC）。

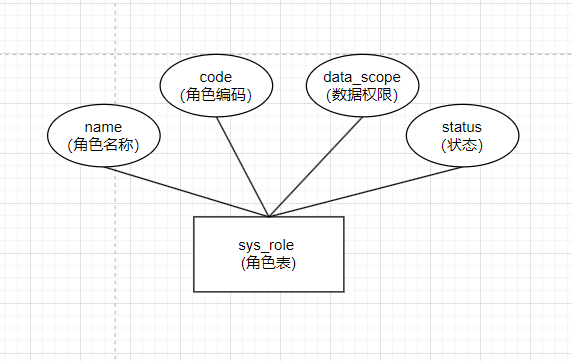


图3.13 角色实体属性图

1. 用户实体（sys\_user）​​

包含用户名、密码、昵称、性别、部门ID、头像、联系方式等属性，实现用户身份认证与基础信息管理。

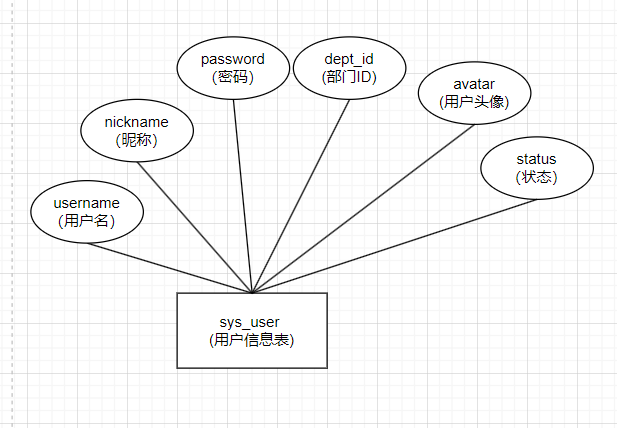


图3.14 用户实体属性图

1. 系统日志实体（sys\_log）​​

包含操作模块、请求参数、IP地址、执行时间、浏览器信息等属性，记录用户操作和系统运行日志。

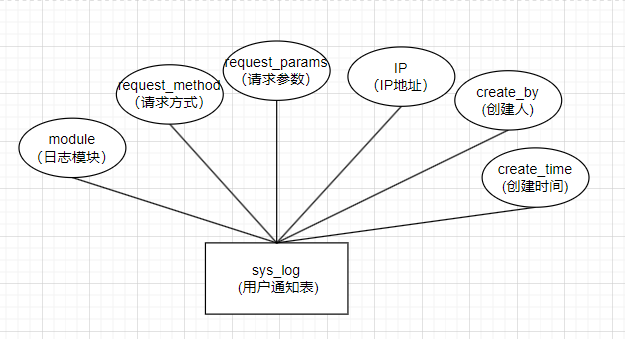


图3.15 用户通知实体属性图

1. 演唱会信息实体（movies）​​

包含演唱会的基本信息，如标题、演出人或团队、演出日期等。主键是id，自增INT类型。

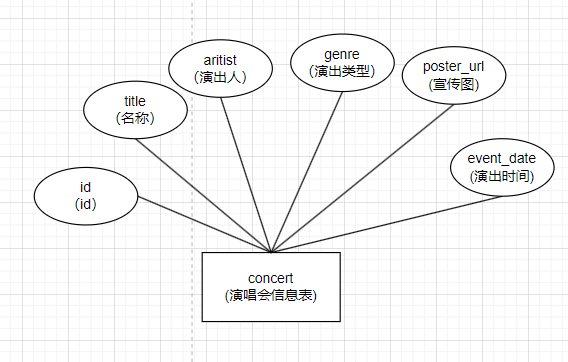


图3.16 演唱会信息实体属性图

1. 演唱会场次实体（movie\_sessions）​​

记录演唱会票的预订信息，包括演唱会、场次、座位、用户等信息，外键关联到其他表。

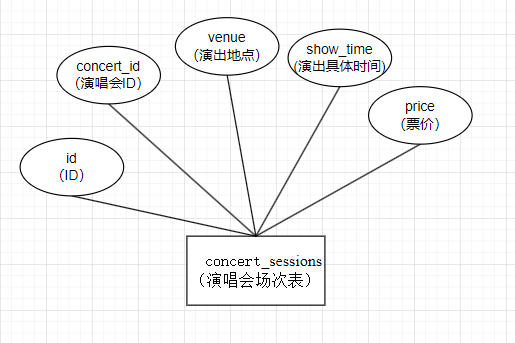


图3.17 演唱会场次实体属性图

1. 座位信息实体（movie\_seats）​​

存储演唱会座位的信息，如场次和座位。

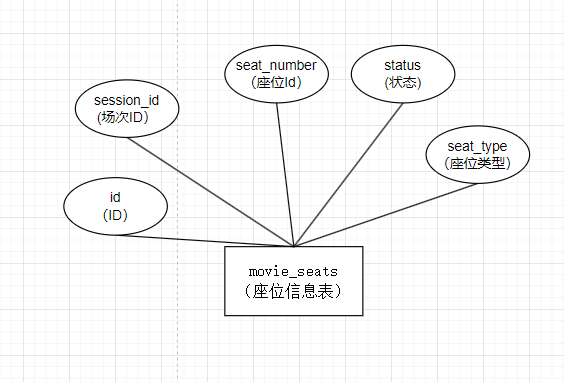


图3.18 座位信息实体属性图

1. 演唱会票实体（movie\_tickets）​​

包含场次ID、座位ID、用户ID等，记录用户下单成功后为用户提供的演唱会票购票信息。

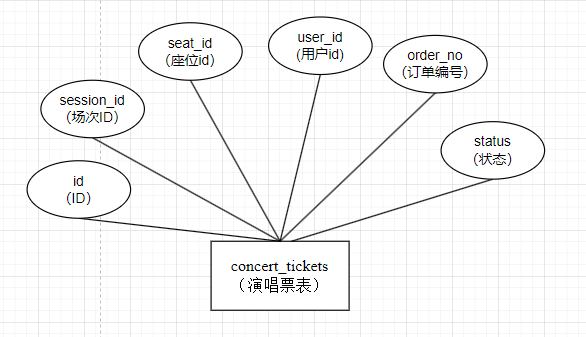


图3.19 演唱会票实体属性图

1. 影评实体（comments）​​

包含演唱会ID、用户ID、内容、用户评分、点赞数等属性，用于用户的评论状态，记录用户信息。

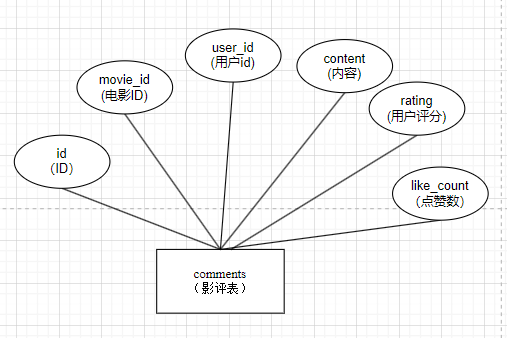


图3.20 影评实体属性图

1. 演唱会分类实体（movie\_categories）​​

包含演唱会信息的类型。

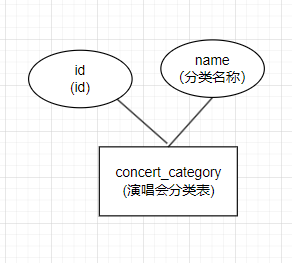
，

图3.21 演唱会分类实体属性图

3.4.2 逻辑结构设计

在此阶段需要将概念设计阶段获取的E-R图（实体-关系图）转化为与所选数据库管理系统兼容的逻辑结构。首先将E-R图转化为相应的关系模式，接着将其适配到数据库管理系统所支持的具体数据模型中。然后进行数据表的构建，并在表结构构建完成后，对它们进行详尽的优化调整，这些调整包括但不限于确立数据间的相互依赖关系，以及对关系模式实施恰当的分解等。以确保数据结构的合理性和高效性。本系统的主要数据库表如表3.4-表3.16所示：

表3.4 sys\_dept（部门表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| name | varchar | 100 | NO |  | 部门名称 |
| code | varchar | 100 | NO |  | 部门编号（唯一） |
| parent\_id | bigint |  | YES | 0 | 父节点ID |
| tree\_path | varchar | 255 | NO |  | 父节点路径 |
| sort | smallint |  | YES | 0 | 显示顺序 |
| status | tinyint |  | YES | 1 | 状态（1-正常 0-禁用） |
| create\_by | bigint |  | YES |  | 创建人ID |
| create\_time | datetime |  | YES |  | 创建时间 |
| update\_by | bigint |  | YES |  | 修改人ID |
| update\_time | datetime |  | YES |  | 更新时间 |
| is\_deleted | tinyint |  | YES | 0 | 逻辑删除标识 |

表3.5 sys\_dict（字典表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| dict\_code | varchar | 50 | YES |  | 字典类型编码 |
| name | varchar | 50 | YES |  | 类型名称 |
| status | tinyint |  | YES | 0 | 状态（0-正常 1-禁用） |
| remark | varchar | 255 | YES |  | 备注 |
| create\_time | datetime |  | YES |  | 创建时间 |

表3.6 sys\_menu（菜单表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| parent\_id | bigint |  | NO |  | 父菜单ID |
| tree\_path | varchar | 255 | YES |  | 父节点路径 |
| name | varchar | 64 | NO |  | 菜单名称 |
| type | tinyint |  | NO |  | 菜单类型（1-菜单 2-目录 3-外链 4-按钮） |
| route\_path | varchar | 128 | YES |  | 路由路径 |
| component | varchar | 128 | YES |  | 组件路径 |
| perm | varchar | 128 | YES |  | 权限标识 |
| visible | tinyint |  | YES | 1 | 显示状态（1-显示 0-隐藏） |
| sort | int |  | YES | 0 | 排序 |
| icon | varchar | 64 | YES |  | 菜单图标 |

表3.7 sys\_role（角色表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| name | varchar | 64 | NO |  | 角色名称（唯一） |
| code | varchar | 32 | NO |  | 角色编码（唯一） |
| data\_scope | tinyint |  | YES |  | 数据权限（0-全部 1-本部门） |
| status | tinyint |  | YES | 1 | 状态（1-正常 0-停用） |

表3.8 sys\_user（用户信息表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| username | varchar | 64 | YES |  | 用户名（唯一） |
| nickname | varchar | 64 | YES |  | 昵称 |
| password | varchar | 100 | YES |  | 密码 |
| dept\_id | int |  | YES |  | 部门ID |
| avatar | varchar | 255 | YES |  | 用户头像 |
| status | tinyint |  | YES | 1 | 状态（1-正常 0-禁用） |

表3.9 sys\_user\_role（用户角色关联表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| user\_id | bigint |  | NO |  | 用户ID（联合主键） |
| role\_id | bigint |  | NO |  | 角色ID（联合主键） |

表3.10 sys\_log（系统日志表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| module | varchar | 50 | NO |  | 日志模块 |
| request\_method | varchar | 64 | NO |  | 请求方式 |
| ip | varchar | 45 | YES |  | IP地址 |
| execution\_time | bigint |  | YES |  | 执行时间（ms） |
| create\_time | datetime |  | YES |  | 创建时间 |

表3.11 sys\_notice（通知公告表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| title | varchar | 50 | YES |  | 通知标题 |
| type | tinyint |  | NO |  | 通知类型（字典notice\_type） |
| publish\_status | tinyint |  | YES | 0 | 发布状态（0-未发布 1-已发布） |
| create\_time | datetime |  | NO |  | 创建时间 |

表3.12 concert（演唱会信息表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| title | varchar | 100 | NO |  | 演唱会名称 |
| artist | varchar | 100 | NO |  | 演出艺人 |
| performers | text |  | YES |  | 参演阵容 |
| poster\_url | varchar | 255 | YES | NULL | 宣传图URL |
| event\_date | datetime |  | NO |  | 演出日期 |
| genre | varchar | 50 | YES |  | 音乐类型 |
| rating | float |  | YES |  | 热度指数 |
| description | text |  | YES |  | 演出描述 |
| create\_by | bigint |  | YES |  | 创建人ID |
| create\_time | datetime |  | YES | CURRENT\_TIMESTAMP | 创建时间 |
| update\_by | bigint |  | YES |  | 修改人ID |
| update\_time | datetime |  | YES | CURRENT\_TIMESTAMP | 更新时间 |
| is\_deleted | tinyint |  | YES | 0 | 逻辑删除标识(0-未删除 1-已删除) |

表3.13 concert\_sessions（演唱会场次表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| concert\_id | bigint |  | NO |  | 演唱会ID |
| venue | varchar | 100 | NO |  | 演出场馆 |
| show\_time | datetime |  | NO |  | 演出时间 |
| price | decimal | 10,2 | NO |  | 票价 |
| status | tinyint |  | YES | 0 | 0-待开始 1-进行中 2-已结束 3-取消 |
| create\_by | bigint |  | YES |  | 创建人ID |
| create\_time | datetime |  | YES | CURRENT\_TIMESTAMP | 创建时间 |
| update\_by | bigint |  | YES |  | 修改人ID |
| update\_time | datetime |  | YES | CURRENT\_TIMESTAMP | 更新时间 |
| is\_deleted | tinyint |  | YES | 0 | 逻辑删除标识 |

表3.14 concert\_seats（座位信息表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 自增主键 |
| session\_id | bigint |  | NO |  | 场次ID |
| area | varchar | 10 | NO |  | 区域代码（如：A、B、VIP） |
| seat\_number | varchar | 10 | NO |  | 座位编号 |
| seat\_type | tinyint |  | YES | 1 | 座位类型(1-普通 2-VIP) |
| status | tinyint |  | YES | 0 | 状态(0-可用 1-已锁定 2-已售出) |
| create\_by | bigint |  | YES |  | 创建人ID |
| create\_time | datetime |  | YES | CURRENT\_TIMESTAMP | 创建时间 |
| update\_by | bigint |  | YES |  | 修改人ID |
| update\_time | datetime |  | YES | CURRENT\_TIMESTAMP | 更新时间 |
| is\_deleted | tinyint |  | YES | 0 | 逻辑删除标识 |

表3.15 concert\_tickets（演唱会票表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| session\_id | bigint |  | NO |  | 场次ID |
| seat\_id | varchar | 20 | NO |  | 座位ID |
| user\_id | bigint |  | NO |  | 用户ID |
| order\_no | varchar | 50 | NO |  | 订单编号 |
| status | tinyint |  | NO | 0 | 状态(0-已预订 1-已支付 2-已取消) |
| create\_by | bigint |  | YES |  | 创建人ID |
| create\_time | datetime |  | YES | CURRENT\_TIMESTAMP | 创建时间 |
| update\_by | bigint |  | YES |  | 修改人ID |
| update\_time | datetime |  | YES | CURRENT\_TIMESTAMP | 更新时间 |
| is\_deleted | tinyint |  | YES | 0 | 逻辑删除标识 |

表3.16 reviews（用户评论表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| concert\_id | bigint |  | NO |  | 演唱会ID |
| user\_id | bigint |  | NO |  | 用户ID |
| content | text |  | NO |  | 评价内容 |
| rating | tinyint |  | YES |  | 用户评分(1-5星) |
| like\_count | int |  | YES | 0 | 点赞数 |
| create\_by | bigint |  | YES |  | 创建人ID |
| create\_time | datetime |  | YES | CURRENT\_TIMESTAMP | 创建时间 |
| update\_by | bigint |  | YES |  | 修改人ID |
| update\_time | datetime |  | YES | CURRENT\_TIMESTAMP | 更新时间 |
| is\_deleted | tinyint |  | YES | 0 | 逻辑删除标识 |

# **4** 系统实现

## 4.1 系统目录结构

该演唱会票务与会员服务平台主要为用户、管理员使用，借助前端界面来与系统实现交互，管理员则通过后端界面来与系统进行连接，从而实现对演唱会票务与会员服务平台各个功能模块的管理。系统目录图如4.1图所示：

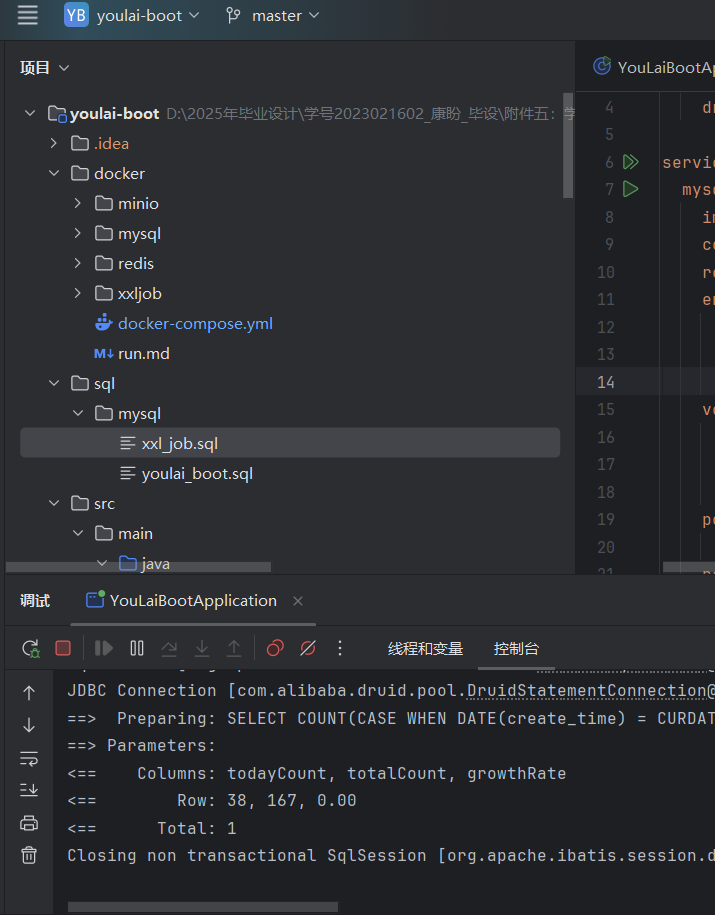


图4.1 系统目录图

## 4.2 系统前端功能模块的实现

4.2.1 前端的主界面模块

用户打开浏览器搜索网址http://localhost:3000/进入演唱会票务与会员服务平台的首页，用户可以点击“登录”进入自己账号的系统首页。首页的设计布局方式基于开源项目youlaiboo，进入系统前端首页，可以查看到首页，系统管理，代码生成等功能模块，并可以进行详细的操作，在首页可以查看到公告。界面如图4.2所示：

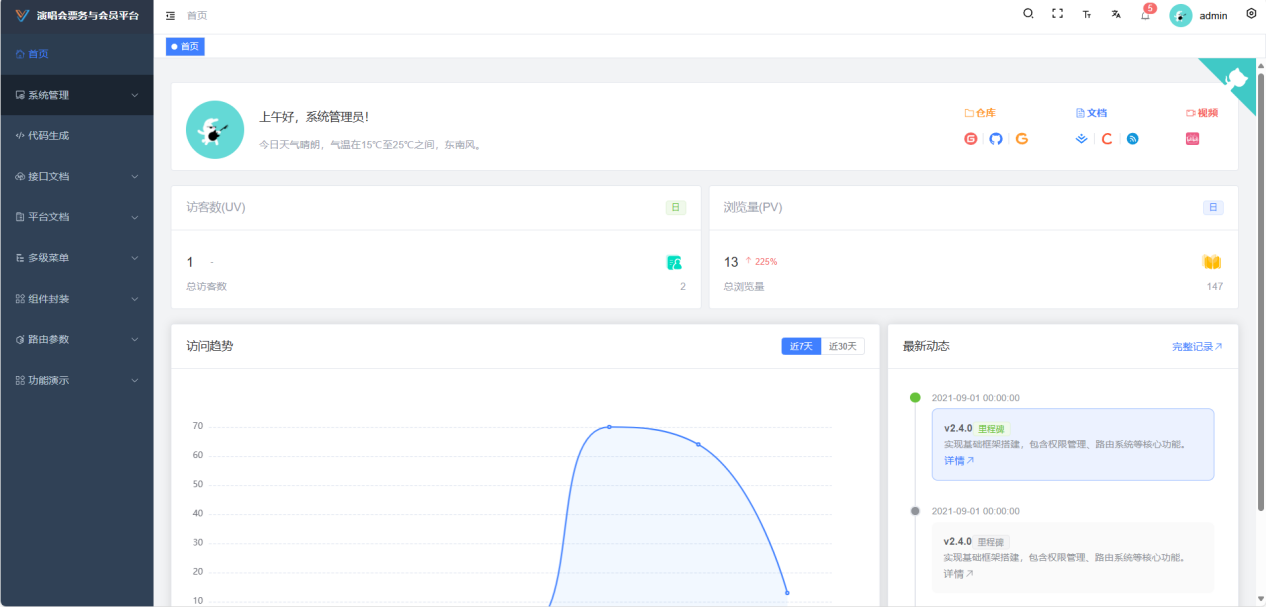


图4.2 系统前端首页

4.2.2 登录模块

当用户使用演唱会票务与会员服务平台时，需要先登录本系统方可进行操作。如果用户尚未注册，必须先完成账号录入功能。在此过程中，用户需点击“注册”按钮，随后按照要求填写包括账号、密码、姓名、性别、联系方式、身份证号和邮箱等在内的个人信息。用户登录流程图如下图4.3所示，界面如图4.4、图4.5所示：

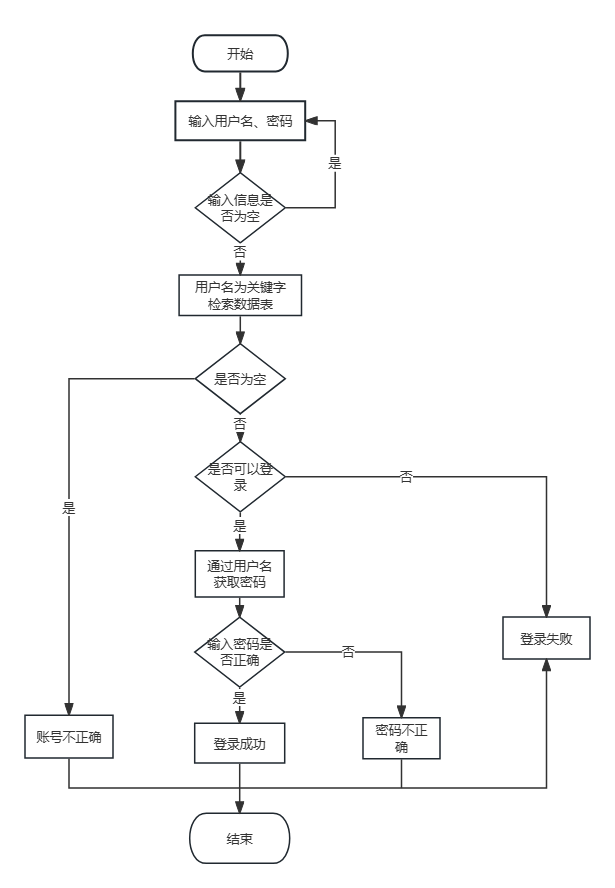


图4.3 用户登录流程图



图4.4 用户登录界面

4.2.3 管理员信息模块

经由RABC权限机制分配权限，管理员进行登录操作后，在UniApp前台首页的演唱会信息模块可以查看到系统所有在移动端前端的详细信息，包含首页轮播图、用户管理、角色管理等。管理员作为系统的最高权限拥有者，可以对系统所有可操作的内容进行操作。点击“用户管理”即可立即进入到用户管理界面，对系统用户进行操作。如图4.6、图4.7、图4.8所示：

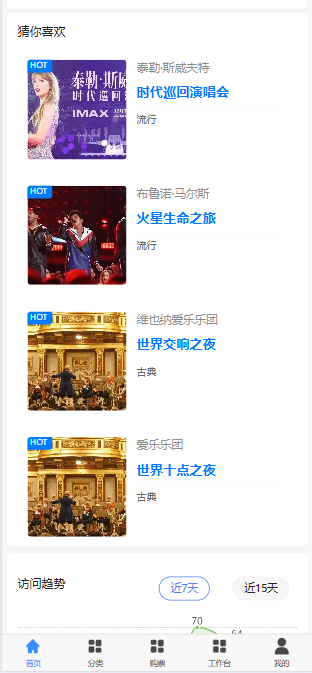


图4.6 首页信息展示界面

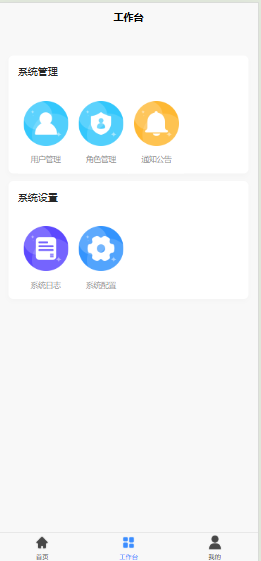


图4.7 系统管理相关

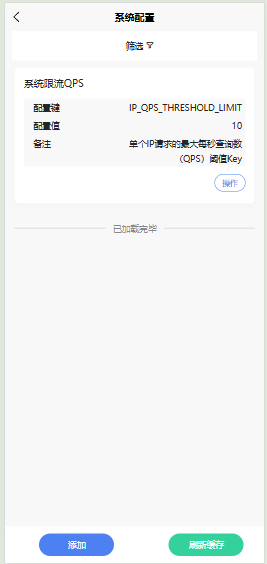


图4.8 系统设置相关

4.2.4 演唱会分类与评价展示模块

在前台首页中并没有管理员权限所拥有的用户管理等功能，但也可以查看到系统相关的用户数据，用户可以左侧分类列表的演唱会分类进行查看对应演唱会分类的演唱会场次；此外用户可以点击查看评价，查看对应演唱会的用户评论。具体如图4.9、图4.10所示：



图4.9 分类展示界面

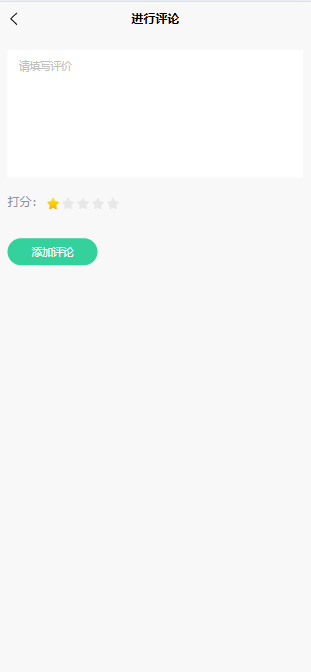
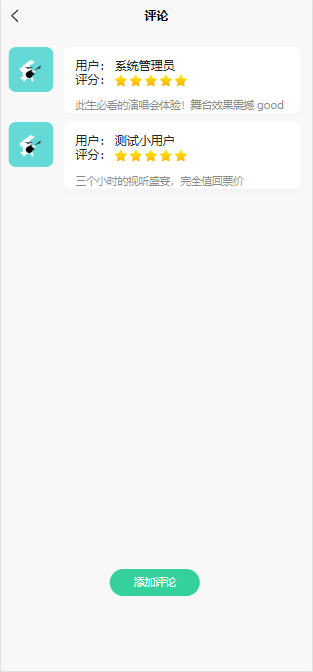


图4.10 评价与添加界面

4.2.5 购票模块

经由RABC权限机制分配权限，所属VIP角色权限的用户，在购票模块可以查看到对应演唱会的信息，并对该些功能模块进行详细的操作，同时，VIP用户是购买的主力单位，主要使用在于购买这个流程，用户可以选择座位进行提交订座位信息，还可以通过支付购买完成订单。具体如图4.12、4.13所示：

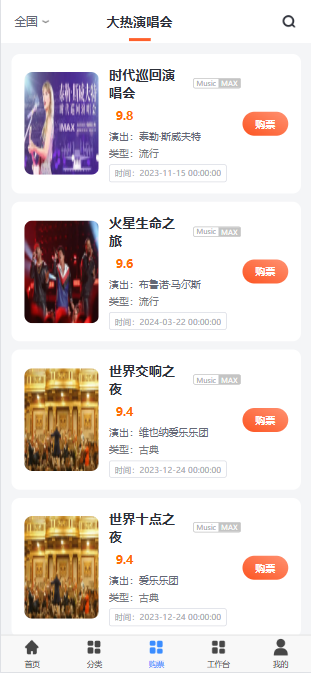


图4.12 购票与指定场次信息

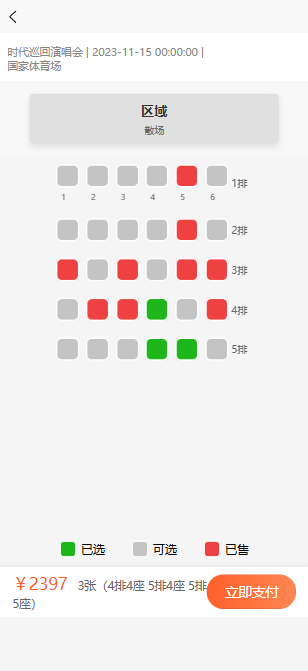


图4.13 支付详情

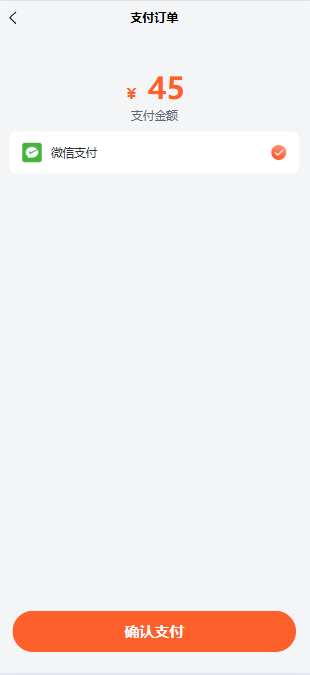


图4.14 支付详情



图4.15 票务详情

4.2.6 个人中心模块

个人中心模块，使用用户来对个人、应用的信息进行查看已经修改的地方，包含个人信息修改、请假记录查询、关于我们等一系列内容。具体如图4.14、图4.15所示：

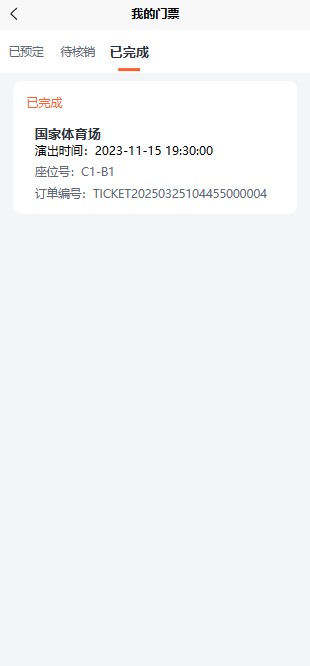


图4.14 票务查看界面

## 4.3 系统后端功能模块的实现

4.3.1 登录模块

管理员登录后台模块进行详细的操作，主要是输入用户名和密码，选择登录身份进行登录操作。登录系统流程图如图4.17所示，相关界面展示如图4.18所示：

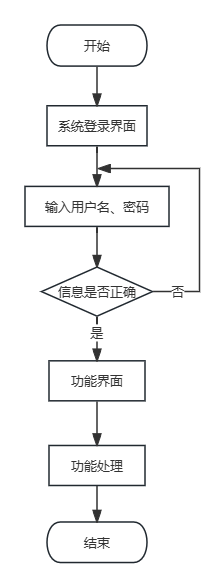


图4.17 管理员、管理员登录流程图



图4.18 登录界面

4.3.2 用户管理功能模块

管理员在用户管理模块可以查看到系统用户所有的姓名，头像，性别，联系方式，等信息，可以对会员信息进行查看详情，修改，删除，重置密码，在该模块可以新增一位用户到系统中来。

点击修改按钮，可以对用户的性别，手机号码等信息进行修改。如图4.19、图4.20所示：



图4.19 用户管理

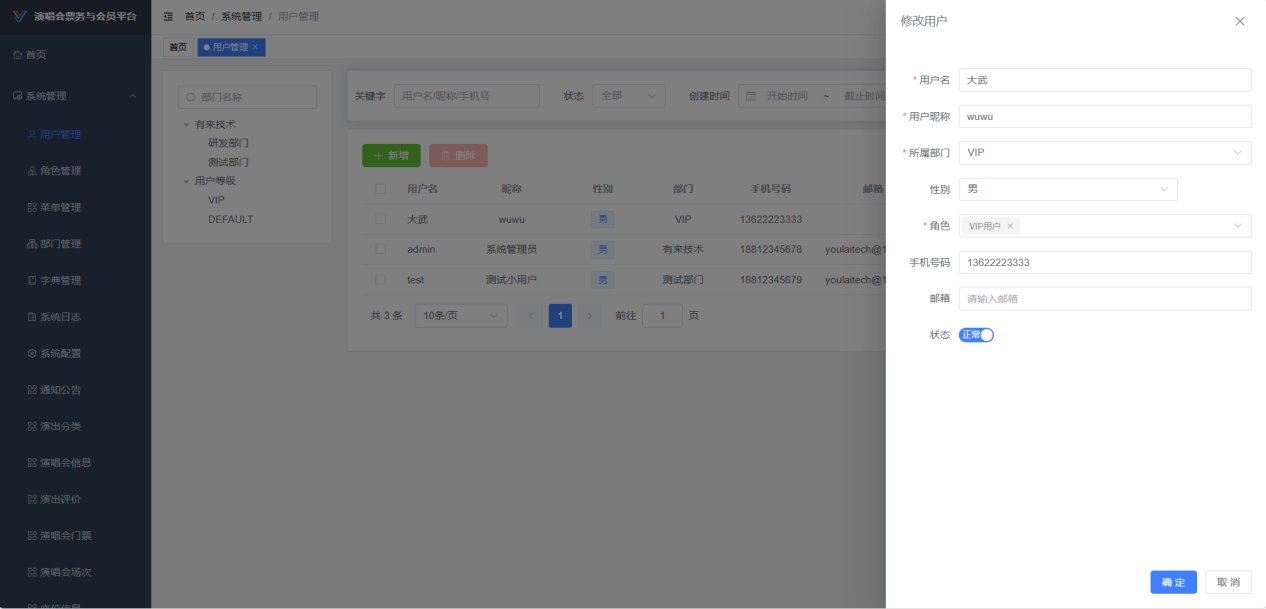


图4.20 修改用户信息界面

4.3.3 角色管理模块

管理员在角色管理模块可以对系统的所有角色详情进行操作，该系统基于RABC权限管理系统，管理员可以为不同的角色设置不同的权限。如下图4.21所示：

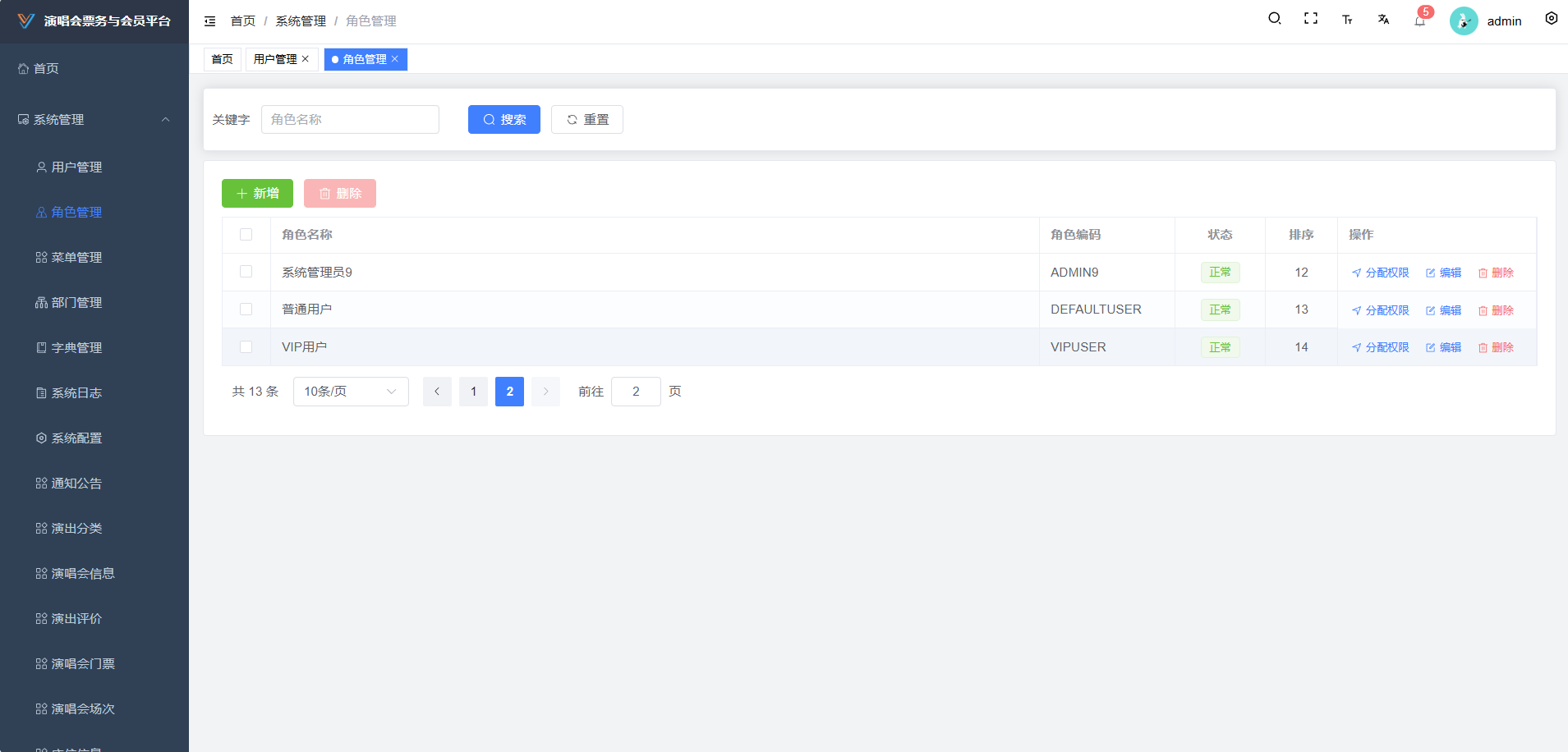


图4.21 角色管理界面

管理员在角色权限分配模块可以对系统所有角色进行查看、操作。如下图4.22所示：

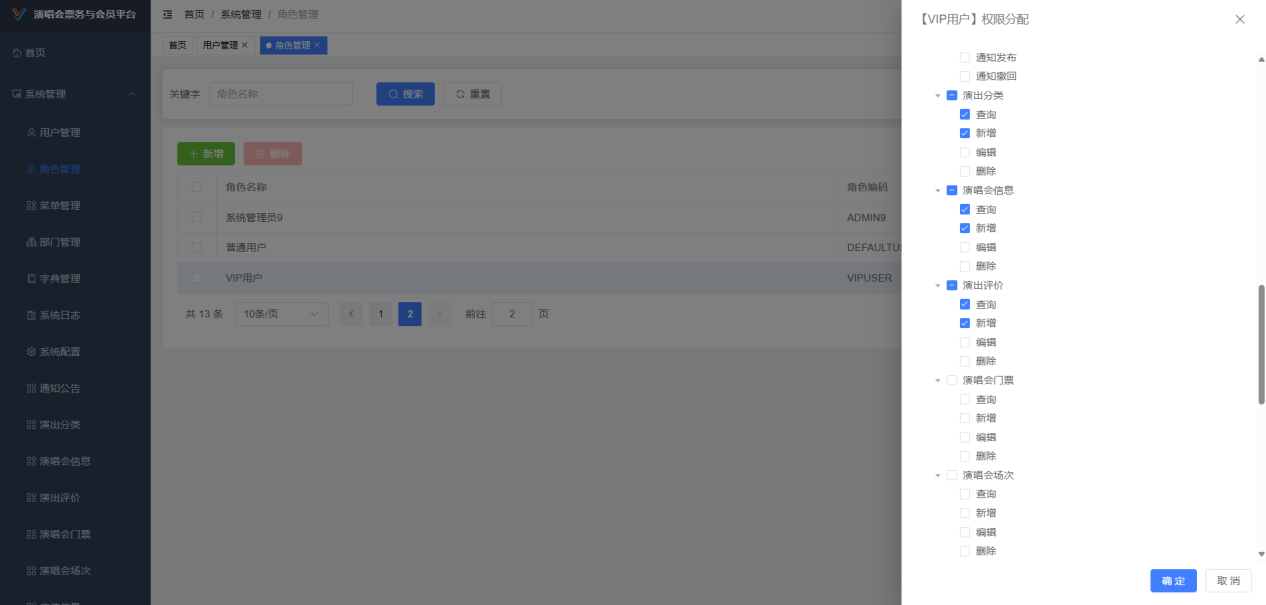


图4.22 角色权限分配界面

4.3.4 部门管理模块

管理员在部门管理模块可以对系统所有部门详细信息进行修改，删除和新增的操作。界面如图4.23所示：

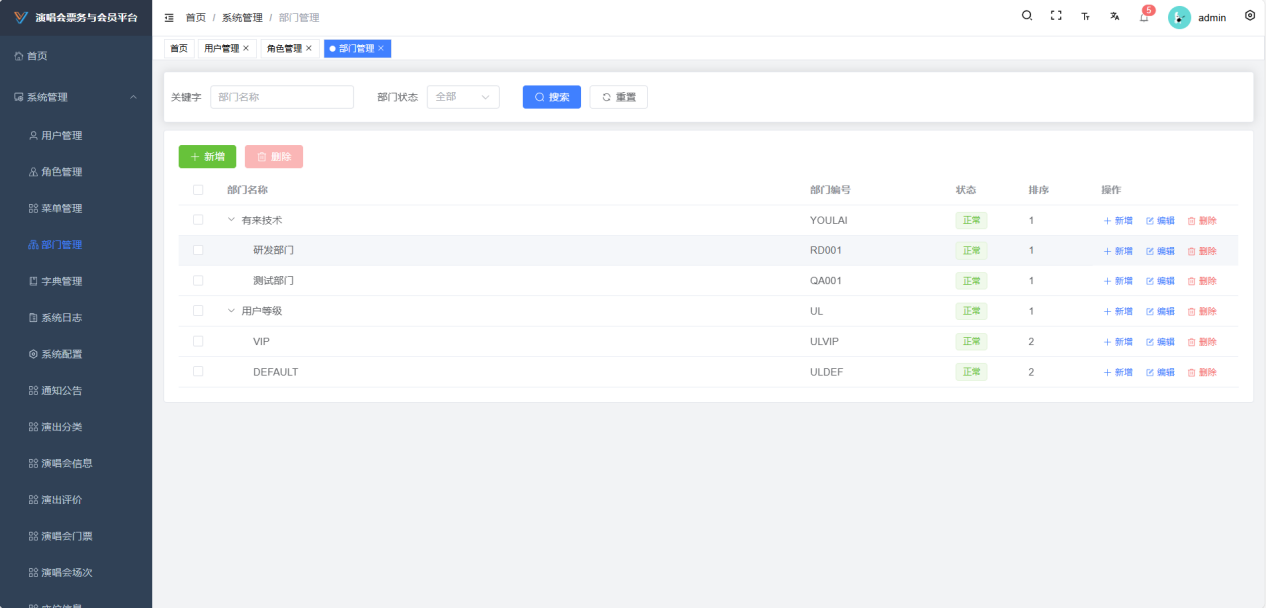


图4.23 部门管理

4.3.5 演出会管理模块

管理员在演唱会信息管理列表可以查看到系统所有的演唱会详情，主要演出时间、演出人、宣传封面、上架时间等信息，可以对对应信息进行修改、删除、新增和查询的操作，在该模块可以对演唱会信息进行有效的修改。。界面如图4.24所示：

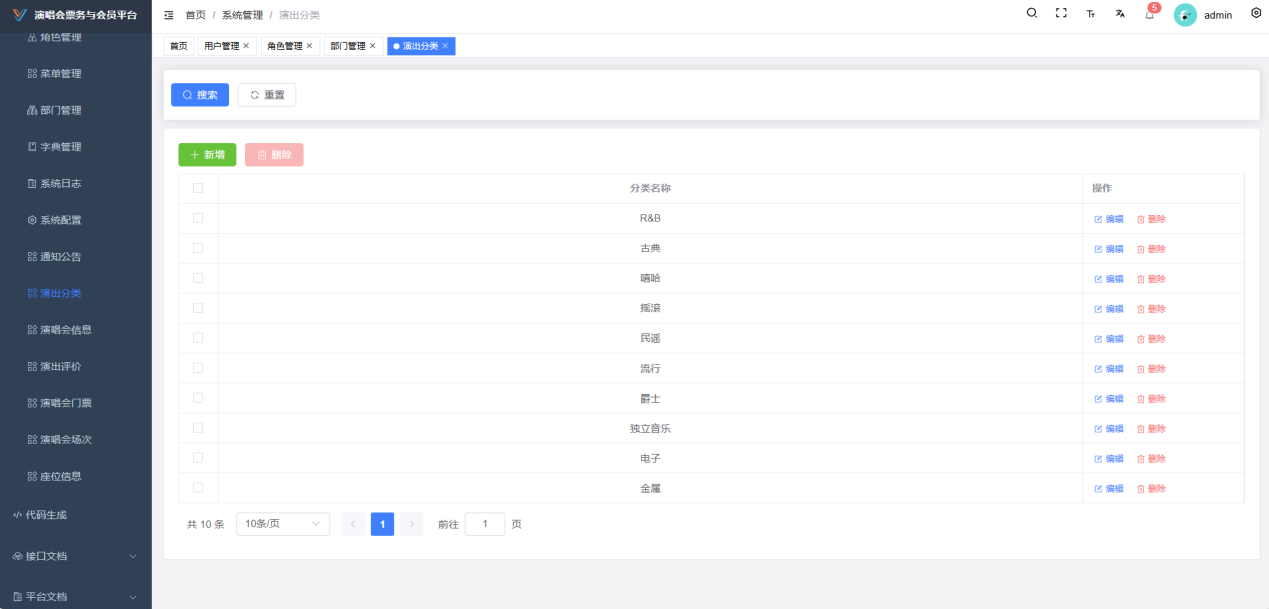
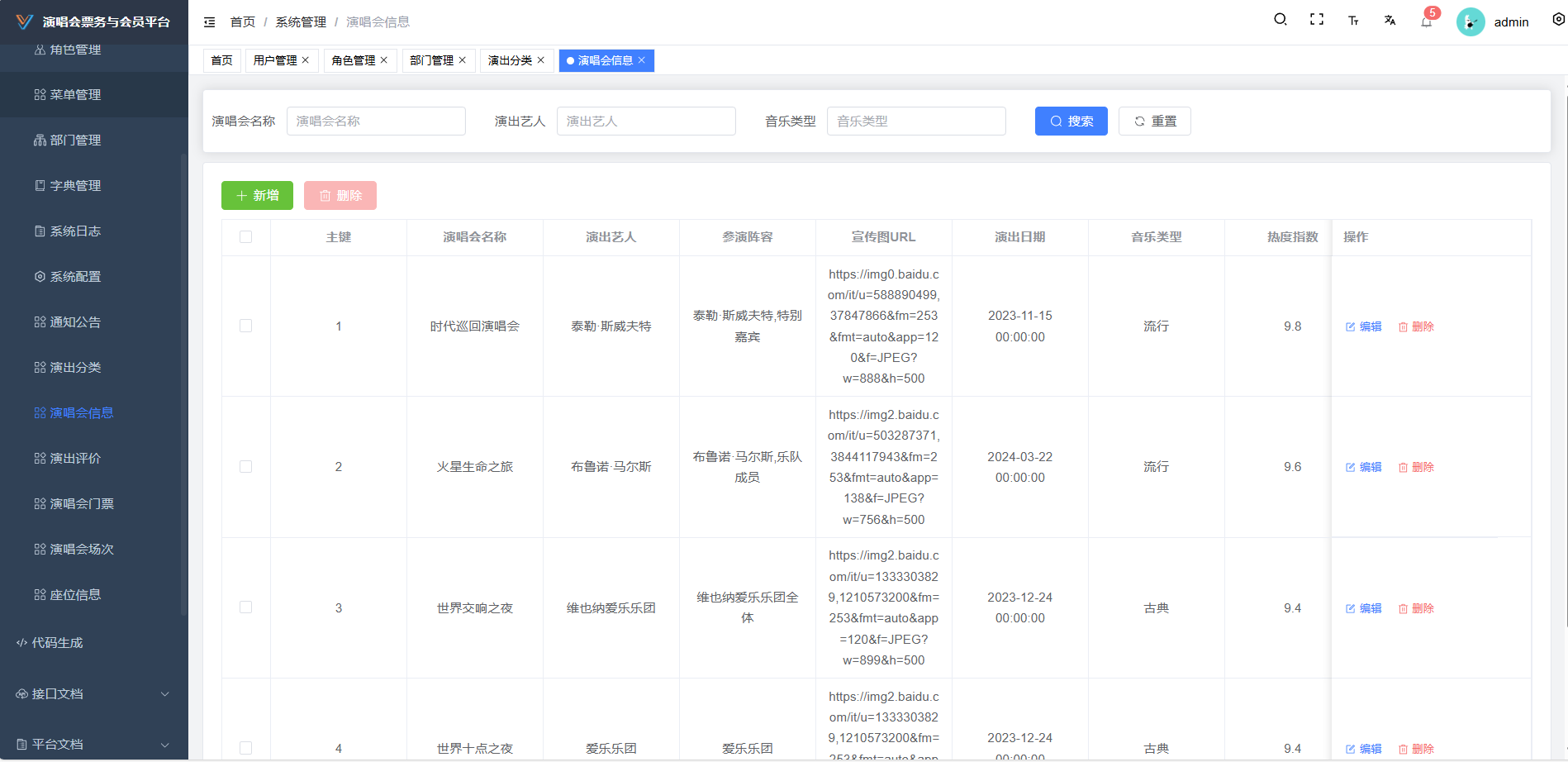


图4.24 演唱会分类信息管理

管理员在该模块可以查看到修改演唱会分类。具体实现界面如图4.25所示：

图4.25 演唱会信息管理

演唱会信息管理，可以对不同的演唱会信息进行增删查改操作，比如特定演唱会的标题修改等。界面如图4.26所示：



图4.26 演唱会评论管理

管理员具有权限可以修改演唱会票务的信息。界面如图4.27所示：

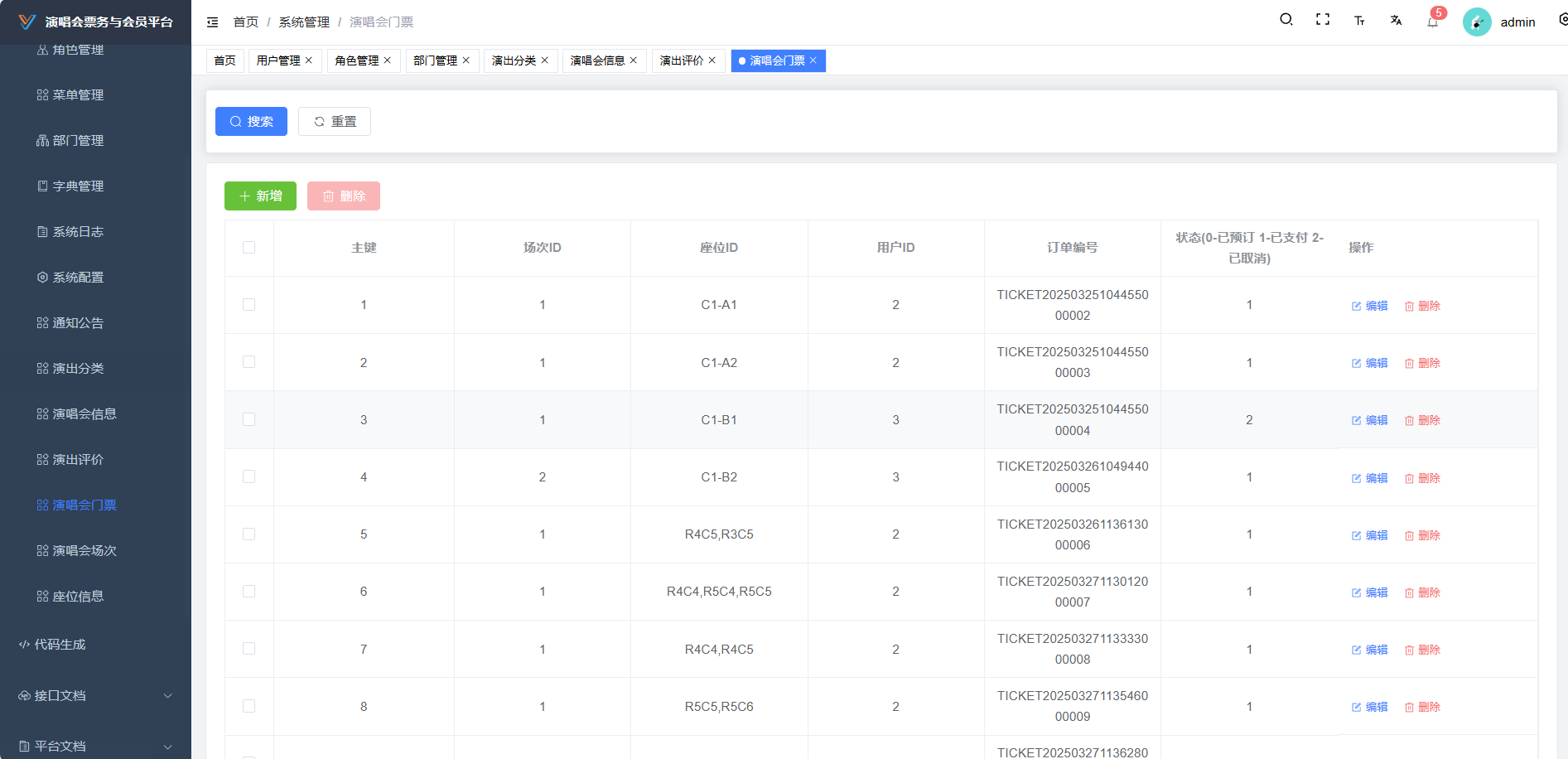


图4.27 演唱会票务管理

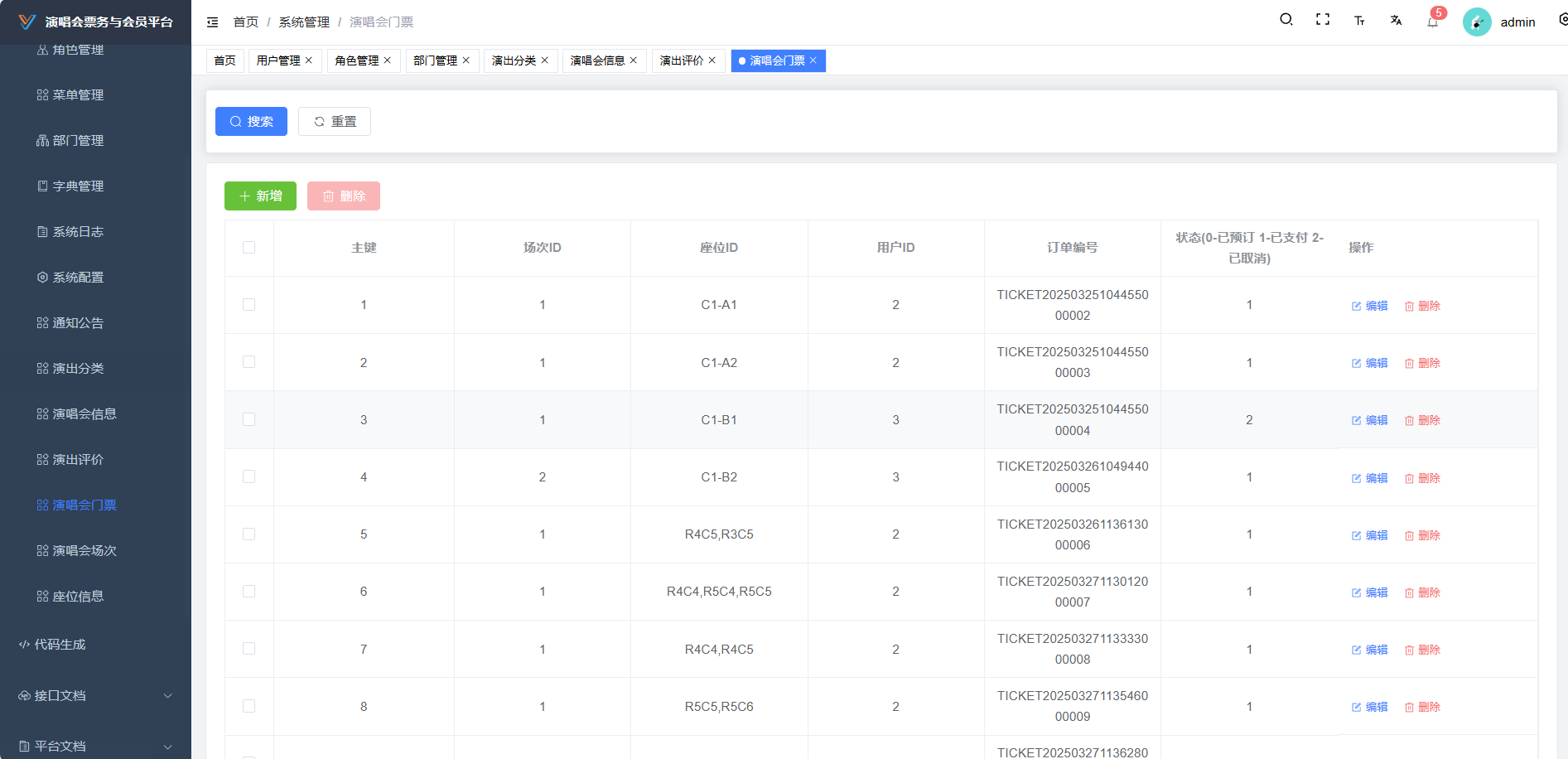
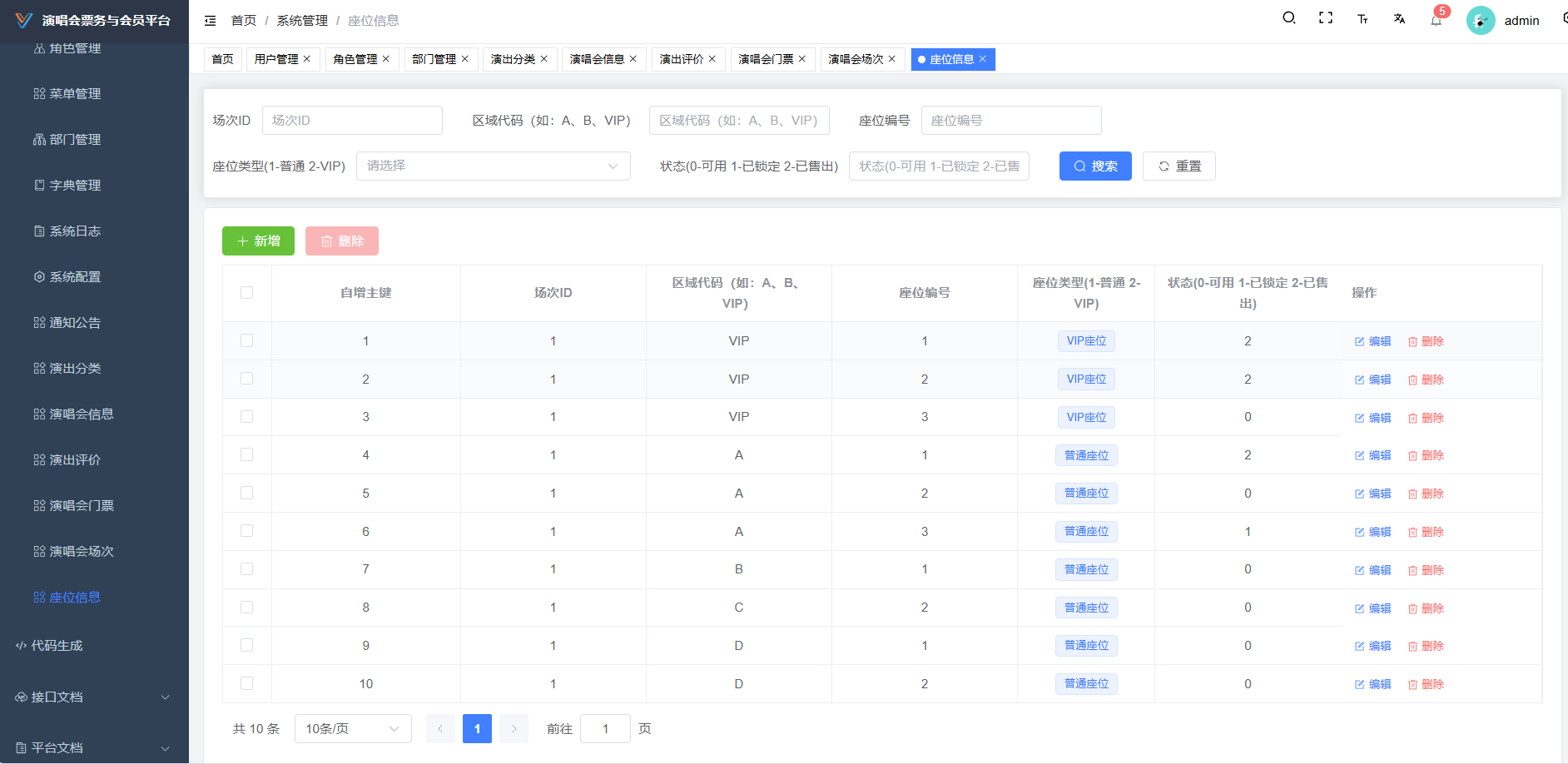


图4.28 演唱会演唱场次管理

 图4.29 VIP座位管理

4.3.6 公告信息管理模块

管理员在公告信息模块可以对系统展示在前台界面的公告详情，公告类型，公告图片和公告时间进行查看详情，修改，删除和新增的操作。具体实现界面如图4.29所示：

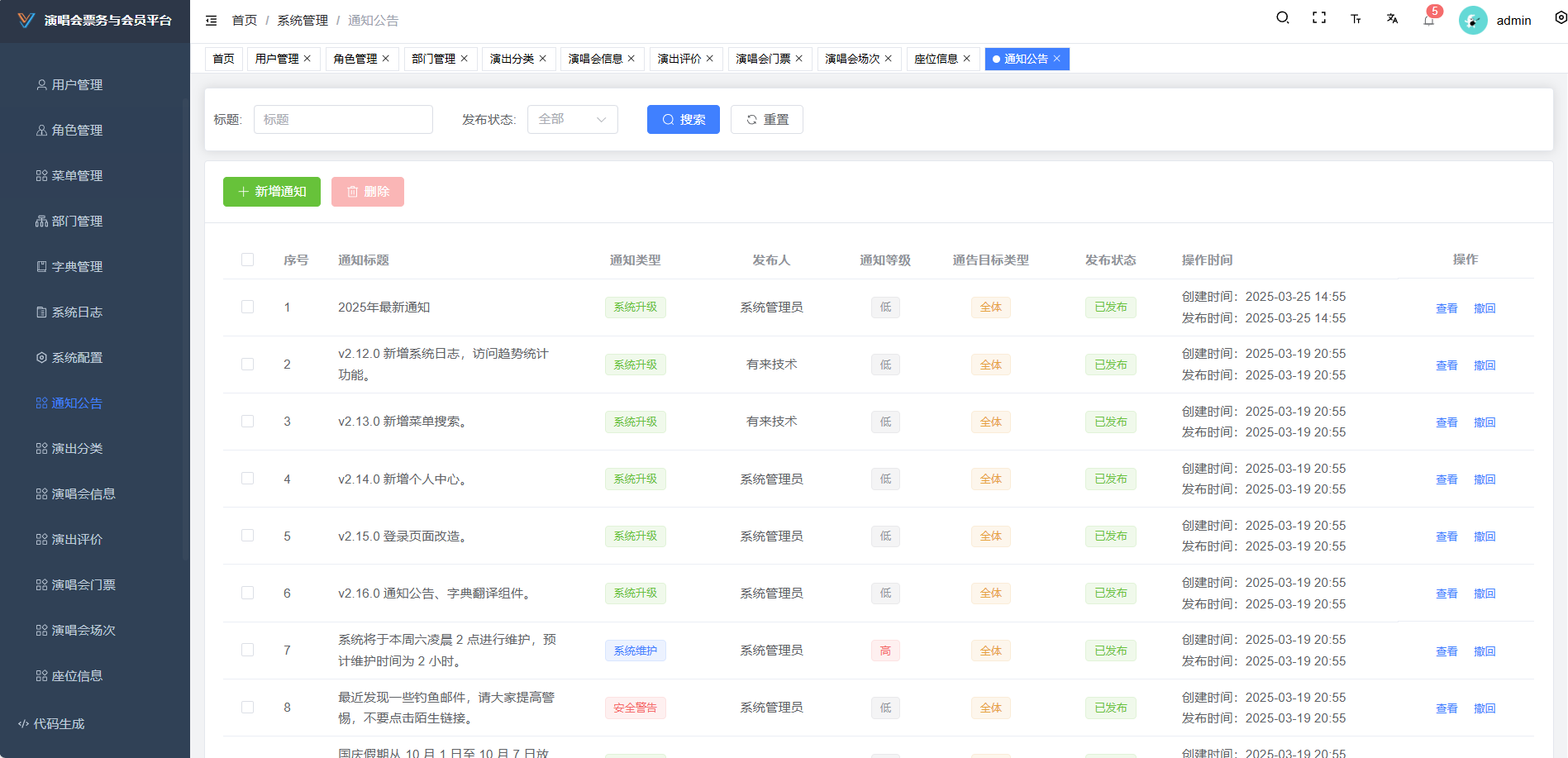


图4.30 管理员公告管理

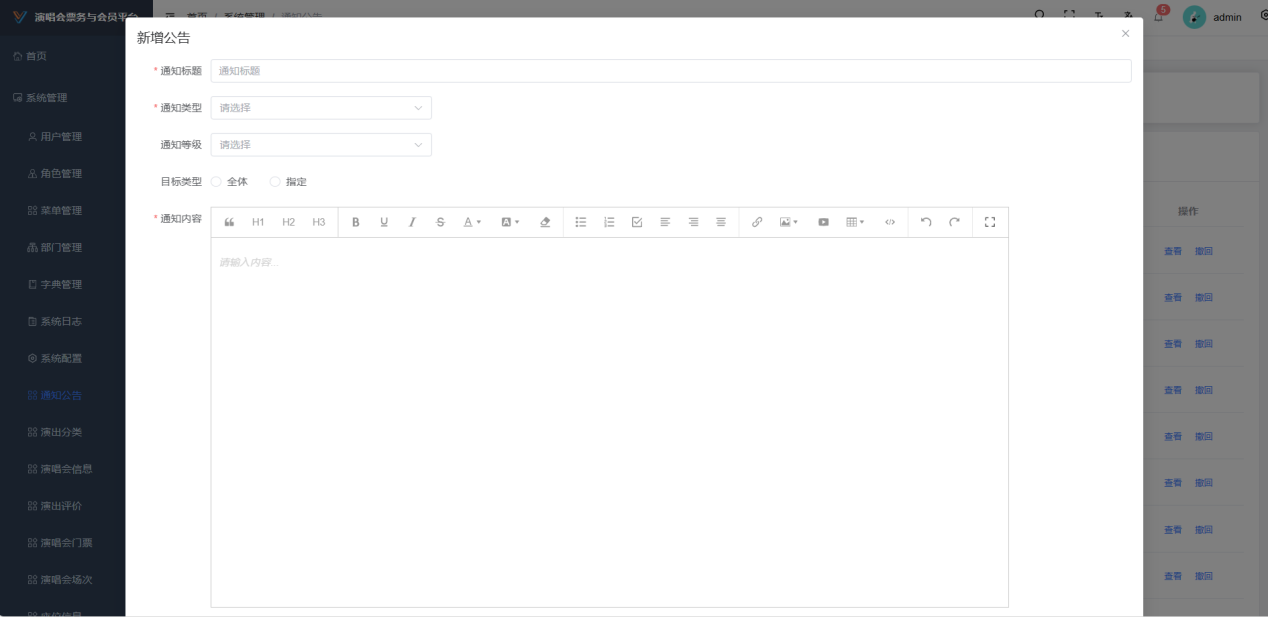
管理员在公告信息模块可以发布系统展示在前台界面的公告详情，界面如图4.30所示： 

图4.31 管理员公告发布

4.3.8 基础字典数据管理模块

管理员在基础数据管理模块可以管理系统展示在前台的字典数据。让高校运营的方向更明确，具体实现界面如图4.33所示：

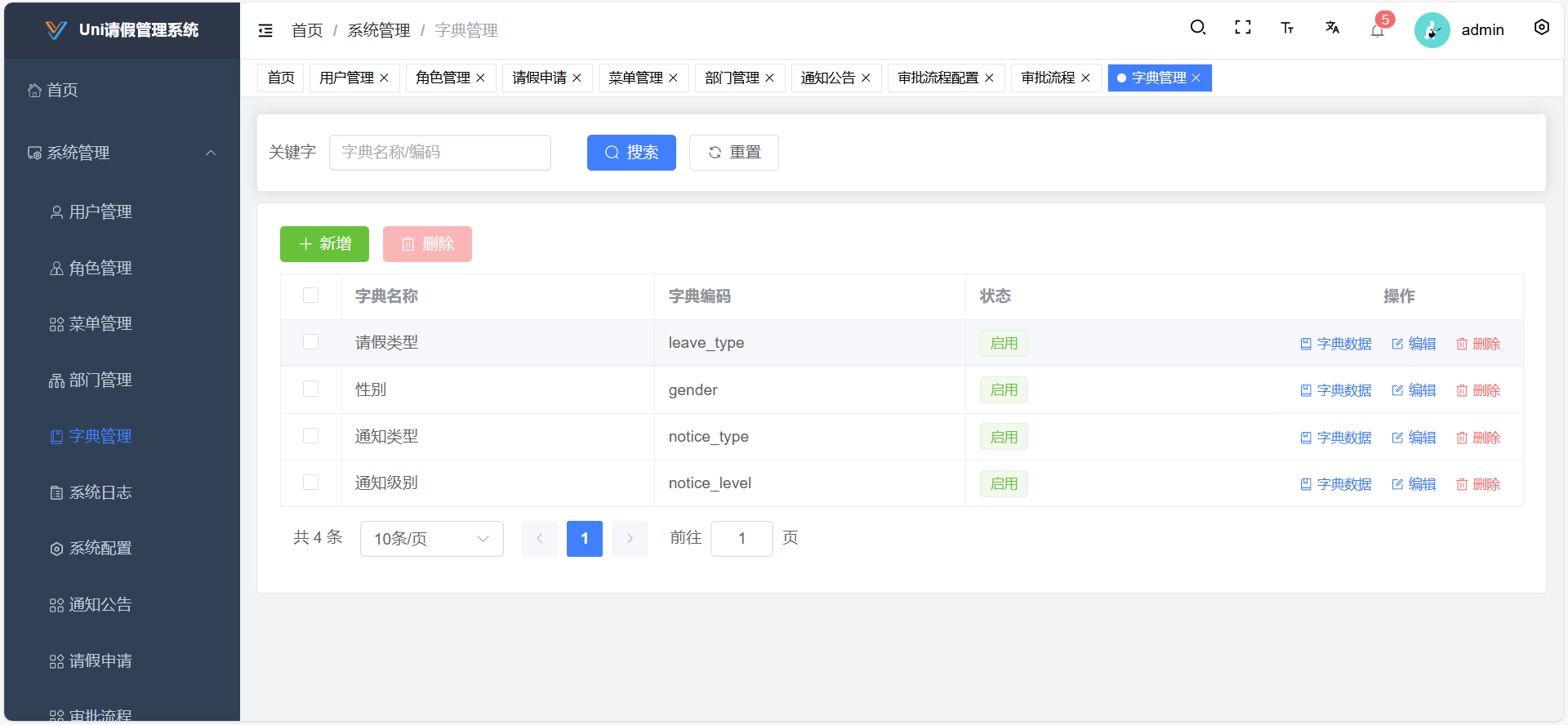


图4.32 基础字典数据管理

4.3.9 系统日志管理模块

管理员在日志管理模块可以管理系统展示在前台的日志。可以让管理员快速直观地了解，并且快速定位到具体问题，具体实现界面如图4.34所示：



图4.33 系统日志管理

# 5 系统测试

当前计算机系统中的软件测试是一个不可或缺的步骤，确保其正常运行是将此类系统推向实用市场的重要环节。经过开发者的充分设计与详细研究后，任何系统在正式投入使用前，均需经历专业的系统测试环节。这一环节不仅是对系统是否满足既定功能规范的验证，更是对系统设计潜在问题的有效识别和优化过程。测试系统的方法很多，本系统主要采用黑盒测试来检查系统的功能和性能表现。

## 5.1 测试的原理

在软件开发中，测试是确保程序正常执行的关键环节。系统的开发目标在于稳定、高效地提供优质服务，确保其在实际应用中不出故障。每个软件在初期阶段都可能存在各种问题，所以需要通过多种测试手段来精准识别系统中的缺陷并逐一解决，以构建出既高效又稳定的系统。这一过程是软件开发中不可或缺的一环。在进行测试时必须遵循既定的规则与准则。这些规则的遵循往往依赖于测试用例的具体要求，而测试用例则是根据详尽的需求规格或说明书精心设计的。本论文选择已实现的功能作为测试的核心要点，从而展开针对性的测试工作。

本论文基于实际的应用场景，选定了以功能为主导的黑盒测试法。此方法适用于系统功能明确的情况下，来验证各功能是否严格遵循需求分析中的定义正常运行。在软件测试中，黑盒测试将程序视为一个封闭系统，无需深入其内部机制。其核心目标在于检验系统的功能需求实现情况，从而发现并修复可能存在的功能缺陷、界面瑕疵以及性能问题等。因此它也常被称为功能验证测试。

## 5.2 系统测试用例

5.2.1 用户注册功能测试

用户录入注册信息后，点击“注册”会出现两种情况：注册成功跳转登录界面或提示请求填写正确的信息。注册功能测试用例表如表5.1所示，注册功能测试结果图如图5.1所示。

表5.1用户注册功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 操作序号 | 操作描述 | 预期输出 | 实际输出 | 结果 |
| 1 | 输入不正确的手机号 | 未注册成功，提示信息为“请重新输入正确的手机号” | 与预期结果相同 | 通过 |
| 2 | 两次输入的密码不一致 | 注册失败，提示信息为“密码的俩次密码不一致” | 与预期结果相同 | 通过 |
| 3 | 输入正确用户名和密码、姓名、手机号、邮箱 | 注册成功，页面跳转到登录界面 | 与预期结果相同 | 通过 |

图5.1　注册功能测试结果图

5.2.2 用户登录功能测试

用户输入登录账号，密码，点击登录按钮，会出现两种情况：登录成功跳转演唱会票务与会员服务平台首页或提示账号或密码不正确。登录功能测试用例表如表5.2所示，登录功能测试结果图如图5.2所示。

表5.2用户登录功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 操作序号 | 操作描述 | 预期输出 | 实际输出 | 结果 |
| 1 | 输入正确用户名和密码 | 登录成功，页面跳转到系统首页 | 与预期结果相同 | 通过 |
| 2 | 输入不正确的密码 | 未登录成功，页面不跳转，提示信息为“账号或密码不正确” | 与预期结果相同 | 通过 |
| 3 | 输入不存在的用户名 | 未登录成功，页面不跳转，提示信息为“账号或密码不正确” | 与预期结果相同 | 通过 |



图5.2　登录功能测试结果图

5.2.3 个人中心功能测试

用户在个人信息管理界面可以对个人资料进行查看修改操作，主要包括用户名、账号、头像、手机号、身份证号、邮箱等详细信息进行修改，查看自己的演唱会订单和请假单兑换记录，对购买的演唱会进行评价或退款。个人中心功能测试用例表如5.3所示，个人中心功能测试结果图如图5.3所示。

表5.3个人中心功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 操作序号 | 操作描述 | 预期输出 | 实际输出 | 结果 |
| 1 | 修改用户名，点击更新信息 | 显示更新后用户名，提示信息为“修改成功” | 与预期结果相同 | 通过 |
| 2 | 进入演唱会订单，选择演唱会点击评价，输入评价并提交 | 返回演唱会订单页面，订单类型由已支付更新为已评价 | 与预期结果相同 | 通过 |
| 3 | 进入演唱会订单，选择演唱会点击退款 | 弹出选项框“确认退款或取消”，确认后返回演唱会订单页面，订单类型由已支付更新为已退款 | 与预期结果相同 | 通过 |

图5.3　个人中心功能测试结果图

5.2.4 演唱会信息管理功能测试

在演唱会信息管理界面中，管理员能够看到系统中所有演唱会的信息以一个列表的形式展现出来，并且还可以对演唱会进行信息的修改，演唱会信息管理测试用例表如表5.4所示，演唱会信息管理测试结果图如图5.4所示。

表5.4 演唱会信息管理模块功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 操作序号 | 操作描述 | 预期输出 | 实际输出 | 结果 |
| 1 | 在管理员页面点击演唱会信息管理 | 跳转到演唱会信息管理页面，以分页列表形式显示所有演唱会信息 | 跳转到演唱会信息管理页面，以分页列表形式显示所有演唱会信息 | 通过 |
| 2 | 在演唱会信息管理页面点击新增，在添加演唱会信息页面输入演唱会信息信息后点击提交 | 成功添加演唱会信息，跳转到演唱会信息管理页面并在演唱会信息列表中显示新增的演唱会信息 | 演唱会信息添加成功，跳转到演唱会信息管理页面并在演唱会信息列表中显示新增的演唱会信息 | 通过 |
| 3 | 在演唱会信息列表中点击修改，在演唱会信息信息页面修改完数据后单击提交 | 成功修改演唱会信息信 息，提示信息为“操作成功”，跳转至演唱会信息管理页面 | 成功修改演唱会信息信息，提示信息为“操作成功”，跳转至演唱会信息管理页面 | 通过 |
| 4 | 在演唱会信息列表中点击删除 | 成功删除演唱会信息，并刷新所有演唱会信息页面 | 成功删除演唱会信息，并刷新所有演唱会信息页面 | 通过 |

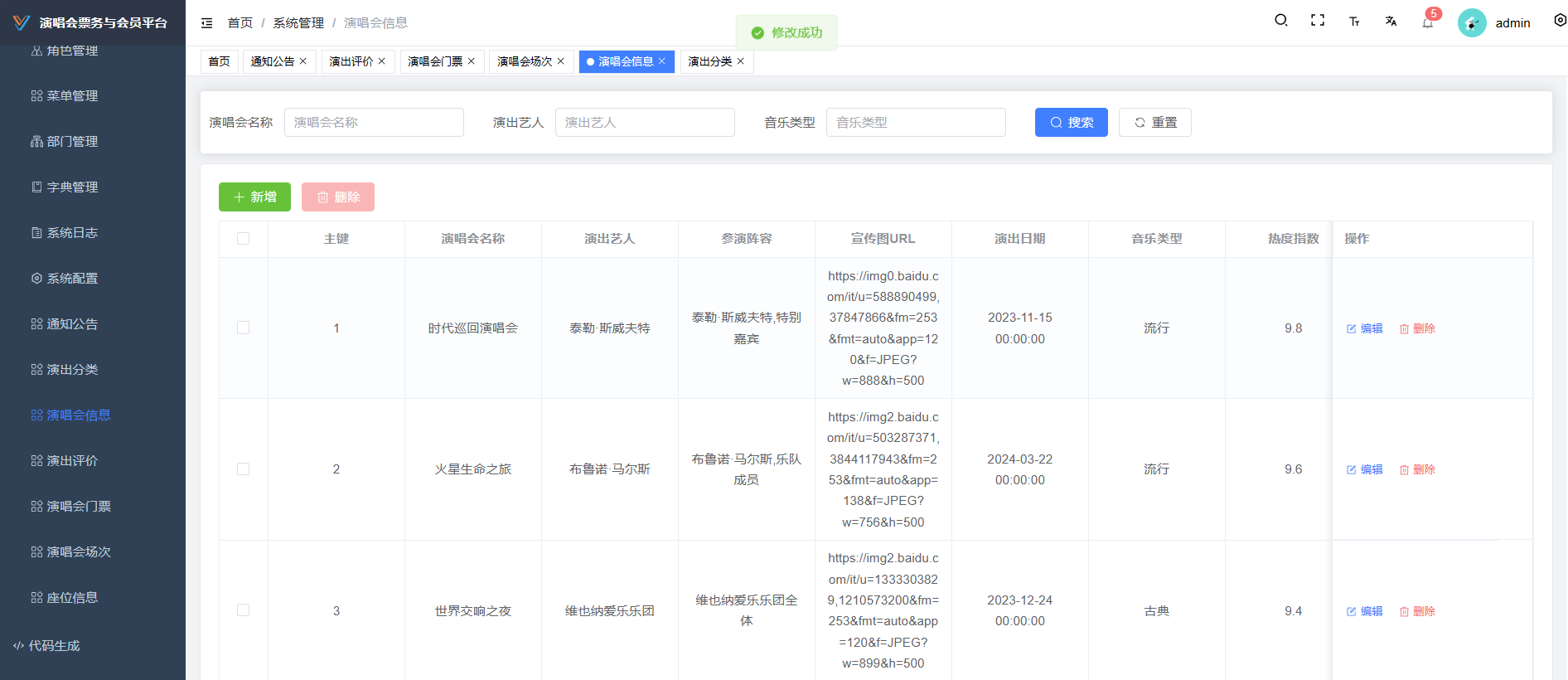


图5.4　演唱会信息管理功能测试结果图

5.2.5 用户管理功能测试

在用户管理界面，系统内的所有用户信息均以列表形式清晰呈现。可以对用户进行信息修改、新增用户或移除用户以及重置密码，确保了用户管理的全面性和灵活性。用户管理测试用例表如表5.5所示。用户管理测试结果图如图5.5所示。

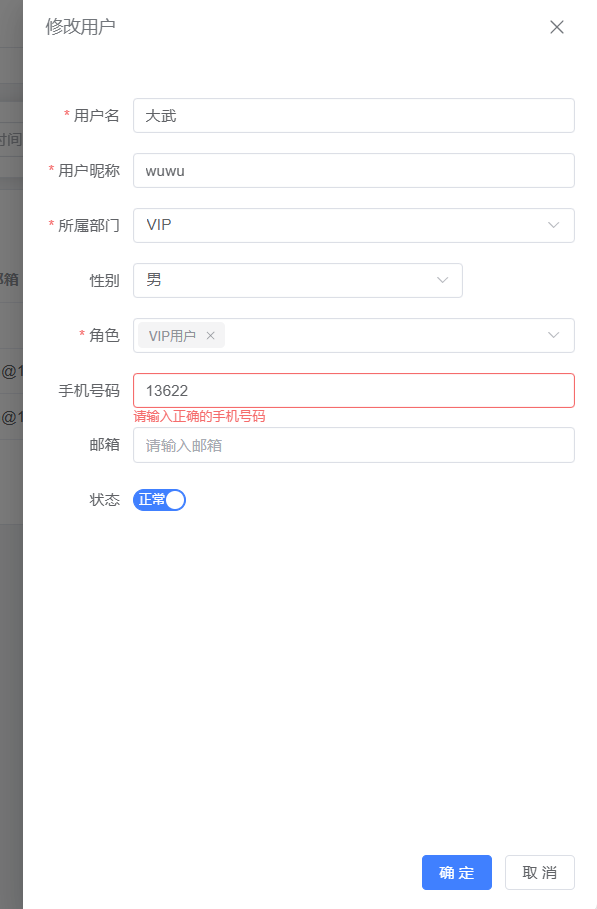


表5.5 用户管理模块功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 操作序号 | 操作描述 | 预期输出 | 实际输出 | 结果 |
| 1 | 在管理员登录页面输入管理员账户和密码，点击登录 | 跳转到用户管理页面，以分页列表形式显示所有用户 | 跳转成功，与预期结果相同 | 通过 |
| 2 | 在用户列表中单击重置密码 | 密码重置成功，提示信息为“重置成功，密码已重置为123456” | 成功重置密码，与预期结果相同 | 通过 |
| 3 | 在用户列表中单击修改，在用户信息页面修改完数据后点击提交 | 成功修改用户信息，提示信息为“操作成功”，更新用户信息列表 | 成功更新用户信息，与预期结果相同 | 通过 |
| 4 | 在用户管理页面点击“新增”，在添加用户页面输入用户信息后单击提交 | 成功添加用户，跳转到用户管理页面并在用户列表中显示新增的用户信息 | 成功添加用户信息，与预期结果相同 | 通过 |



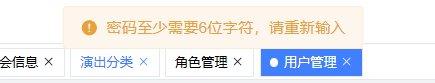


图5.5　用户管理功能测试结果图

## 5.3 测试结果分析

经过以上测试后对演唱会票务与会员服务平台的运行性能进行了分析。系统表现良好，不仅支持后台数据的增删改查操作，而且大部分功能均已成功实现。还具有一定的安全性保障，符合设计的各项要求。

## 5.4 本章小结

系统测试是完善系统必不可少的环节，采用常规的黑盒测试对系统各个功能模块进行了测试，不仅设计了多个测试用例，还在测试过程中严格要求系统是否符合需求。经过不断的测试与调整，使系统达到需求条件。