****

**毕 业 设 计**

**(非定稿)**

**题 目 会议活动票务预定与签到管理系统的设计和实现**

**姓 名**

**学 号**

**学 院**

**专业年级**

**指导教师**

**2025 年 5月 20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：

2025年 5月 20日

**摘 要**

随着企业规模扩大和会议需求增加，传统的手工会议预定方式已难以满足高效管理的需求。为此，本文设计并实现了一套基于现代化技术栈的会议预定管理系统，旨在提升会议预定效率、优化会议活动资源使用，并改善用户体验。

系统采用前后端分离架构，前端使用 Flutter和Vue3 分别实现移动端应用和Web后台管理页面，后端基于 go语言的 Gin框架开发，数据库选用MySQL 存储数据，并通过 GORM 提供类型安全的数据访问能力。同时，Redis 被用作缓存数据库，以提高系统响应速度和并发处理能力。

功能模块上，系统包含用户管理、会议活动管理、会议预定、通知提醒和统计分析五大核心部分。用户管理模块支持注册、登录、权限分配及角色管理；会议活动管理模块提供会议活动的增删改查功能；会议预定模块允许用户选择会议活动、设定时间、填写主题及参会人员信息，系统会自动检测会议活动可用性以避免冲突；通知提醒模块可通过邮件或短信发送预定成功或会议即将开始等通知；统计分析模块则对预定数据进行分析，为资源优化提供决策依据。

在安全性方面，系统通过双Token 实现用户身份验证，确保接口安全。经过测试，该系统运行稳定，能够满足企业会议预定的基本需求，显著提升了预定效率和资源利用率。未来，我们将进一步完善功能，探索与其他企业应用的集成，推动智能化会议管理的发展。

**关键词**  会议活动票务预定；签到管理；MySQL数据库；Go语言

**Title**  **Design and Implementation of a Meeting and Event Ticket Reservation and Check-in Management System**

**Abstract:**

With the increasing scale of enterprises and meeting demands, traditional manual meeting reservation methods have become inefficient and incapable of meeting modern requirements. This paper designs and implements a modern meeting reservation management system using advanced technologies to enhance efficiency, optimize resource utilization, and improve user experience. The system adopts a front-end and back-end separation architecture, utilizing Flutter for the mobile application to provide an intuitive user interface and Vue3 for the web-based management platform to support administrative tasks. On the server side, the Go language's Gin framework is employed for high-performance API development. MySQL serves as the relational database with GORM ensuring type-safe data access, while Redis is integrated as a caching layer to improve response speed and handle high concurrency effectively.

Security is ensured through a dual-token mechanism for robust user authentication, protecting sensitive data and ensuring system integrity. Future work will focus on further functionality improvements with other enterprise applications to promote intelligent meeting management. This system not only streamlines meeting organization but also provides valuable insights for better decision-making, contributing to more efficient enterprise operations.

**Keywords:** Meeting reservation; Resource management; MySQL; Go language; Gin framework; Flutter; Vue3; Redis; Dual-token authentication

[参考文献 45](#_Toc12110)

目 录

[1 绪论 1](#_Toc29901)

[1.1 课题研究背景 1](#_Toc14573)

[1.2 课题研究现状 2](#_Toc30226)

[1.3 本文主要工作 2](#_Toc11521)

[1.4 本文组织结构 3](#_Toc23626)

[2 技术简介 1](#_Toc13470)

[2.1 B/S架构和C/S架构 1](#_Toc4552)

[2.2 前端框架Flutter和Vue 1](#_Toc15819)

[2.3 后端框架Gin 2](#_Toc19822)

[2.4 数据库MySQL 2](#_Toc2291)

[2.6 分布式存储Redis 3](#_Toc26307)

[2.7 开发工具VS Code 4](#_Toc27941)

[3 系统分析与设计 5](#_Toc18362)

[3.1 需求分析 5](#_Toc18233)

[3.2 系统可行性分析 13](#_Toc20915)

[3.3 系统数据流图 14](#_Toc13140)

[3.4 数据库设计 14](#_Toc30585)

[4 系统实现 19](#_Toc27489)

[4.1 系统目录结构 19](#_Toc7248)

[4.2 系统前端功能模块的实现 19](#_Toc14969)

[4.3 系统后端功能模块的实现 27](#_Toc2128)

[5 系统测试 37](#_Toc13385)

[5.1 测试的原理 37](#_Toc1172)

[5.2 系统测试用例 37](#_Toc6398)

[5.3 测试结果分析 43](#_Toc1557)

[5.4 本章小结 43](#_Toc20798)

[总 结 44](#_Toc17447)

[致 谢 45](#_Toc2867)

[参考文献 46](#_Toc16356)

# 

# **1** 绪论

## **1.1** 课题研究背景

在在当今社会，随着经济的快速发展和人们生活水平的提高，各类会议、展览、演出等活动的需求日益增长,同时需要针对系统进行国产化适配[1]。这些活动不仅为企业和个人提供了交流与合作的平台，也成为推动行业发展和文化传播的重要途径。然而，传统的票务预定与签到管理方式主要依赖人工操作，效率低下且容易出错，已无法满足现代活动组织对高效、精准管理的需求。因此，开发一套智能化的会议活动票务预定与签到管理系统显得尤为重要。

传统管理模式存在诸多问题，例如：缺乏线上化的预定渠道，用户需通过线下方式进行票务购买或注册，流程繁琐且耗时；信息传递不畅，活动主办方难以及时获取参与者的详细数据；签到环节效率低下，手工签到容易出现排队拥挤、信息记录错误等问题。这些问题不仅影响了用户体验，还增加了活动组织方的管理成本和运营压力。为解决这些问题，现代活动管理逐渐向数字化、智能化方向转型，通过引入信息技术提升管理效率和服务质量[2]。

随着互联网技术、移动通信技术和大数据分析的快速发展，为会议活动票务预定与签到管理系统的实现提供了坚实的技术基础。借助这些技术，系统可以实现在线票务预定、电子化签到、实时数据统计等功能，极大简化了活动管理流程。例如，通过二维码或人脸识别技术实现快速签到，利用数据分析功能优化活动安排，为参与者提供更加便捷的服务体验。

此外，在市场竞争日益激烈的背景下，活动主办方需要通过创新服务模式来吸引和留住客户。通过引入智能化管理系统，可以根据用户的历史行为和偏好，提供个性化的推荐服务，如定制化的活动套餐或专属优惠，从而提升用户满意度和忠诚度。同时，系统还可以帮助主办方更好地掌握活动数据，为后续决策提供科学依据。

综上所述，设计和实现一套高效的会议活动票务预定与签到管理系统，对于提升活动管理水平、优化用户体验和提高经营效益具有重要的研究意义和实践价值。通过系统化的管理和智能化的服务，不仅可以降低运营成本，还能为参与者带来更加便捷、高效的活动体验，进一步推动相关行业的健康发展。

## **1.2** 课题研究现状

随着社会经济的快速发展，各类会议、展览、演出等线下活动需求持续增长。然而，国内会议活动票务预定与签到管理领域仍存在效率低下、功能不完善的问题，影响了主办方的运营效率和参与者的体验感，制约了行业发展。因此，构建高效、智能的管理系统显得尤为重要。此类系统不仅能解决传统管理模式中的痛点，还能为行业带来新的发展机遇。

国外已进入高度智能化阶段，SaaS模式的管理系统被广泛采用[3]，覆盖从票务预定到现场签到全流程，具备在线预定、电子支付、数据统计等功能，支持云端部署并可随时更新。但这类系统订购成本较高，对中小型主办方存在一定经济压力。在国内，早期传统软件因功能性不足逐渐被淘汰，近年来新兴平台如大麦网、猫眼演出等结合移动互联网技术，提供在线预定、二维码签到等功能，助力数字化转型，但仍存在功能单一、安全性不足及操作复杂等问题，难以完全满足多样化的需求[4]。

未来，会议活动管理系统将呈现智能化升级趋势[5]，通过人工智能实现个性化推荐；支持多终端访问，满足不同用户需求；利用大数据分析优化活动安排；同时加强数据安全防护，确保用户隐私安全。这些发展方向将进一步提升系统性能和服务质量，推动行业的精细化管理和用户体验的全面提升[6]。

综上所述，开发高效的会议活动票务管理系统具有重要意义，不仅可提高管理效率和用户体验，还将推动行业可持续发展。随着技术进步，这一领域有望迎来更多创新，为用户提供更优质的活动服务体验。此外，系统的不断优化也将促进资源的有效配置，助力企业降本增效，为行业的长期健康发展奠定坚实基础[7]。

## 1.3 本文主要工作

本项目可划分为两个主要部分进行：

首先，进行系统的开发与实现工作。系统采用B/S架构，使用Golang语言结合Gin Web框架进行开发，利用其高效、轻量的特点构建高性能的后端服务。开发工具选用支持Golang的现代化集成开发环境VS Code，以提升开发效率。在数据存储方面，选择MySQL作为数据库，因其强大的功能和高可靠性，能够很好地满足项目对数据管理的需求。同时，为确保项目的顺利实施，我们选用了当前流行且成熟的技术栈，以保障系统的性能与扩展性。

其次，进行论文文档的编写工作。需要深入研究课题背景，并通过社会调查全面了解项目实际需求，明确系统的目标与定位。在此基础上，对系统进行详细的设计与分析，包括功能模块划分、技术选型和系统架构设计等内容。随后，展示系统的各项功能实现，并进行全面的功能测试与性能测试，确保系统的稳定性和可靠性。最后，对整个项目进行总结，提炼开发过程中的经验和教训，为后续类似项目的开展提供参考。

通过以上两部分工作的有序开展，项目将实现从需求调研到系统实现再到成果总结的完整闭环，确保最终交付一个高质量的会议活动票务预定与签到管理系统。

## 1.4 本文组织结构

本论文在第一章绪论中，首先探讨了会议活动票务预定与签到管理系统的研究背景和当前现状，明确了系统开发的必要性和可行性，并对论文的整体结构和研究内容进行了概述。

第二章技术简介部分，详细介绍了开发本系统所使用的核心技术。系统采用B/S架构，前端基于Flutter框架开发移动端应用，以实现便捷的移动化操作；同时，基于Vue框架开发后台管理系统[8]，用于满足管理端的功能需求。后端则采用Golang语言结合Gin Web框架构建高效、稳定的服务器端逻辑。此外，数据库选用MySQL，以支持系统的数据存储与管理需求。

第三章系统分析与设计部分，对项目进行了全面的需求分析，包括功能需求和非功能需求，并对系统的可行性进行了详细论证。在此基础上，本章还深入阐述了系统的总体设计思路，涵盖了模块划分、功能设计以及交互流程等内容。同时，针对数据库设计进行了详细介绍，包括概念结构设计、逻辑结构设计以及具体数据库表的设计方案。

第四章系统实现部分，详细描述了系统的开发过程，包括前端界面的设计与实现（移动端和后台管理端）以及后端代码的具体实现。重点展示了关键功能模块的实现细节和技术难点的解决方法。

第五章系统测试部分，对系统进行了全面的功能测试、性能测试及整体集成测试，确保各功能模块的正常运行和系统的稳定性。此外，还进行了验收测试，验证系统是否完全满足用户需求并达到预期目标。

最后，在第六章总结与展望部分，对本论文的研究工作进行了全面总结，回顾了系统开发的主要成果，并提出了未来可能的改进方向和发展前景，为后续研究提供了参考和启发。

# **2** 技术简介

## 2.1 B/S架构和C/S架构

B/S（Browser/Server）架构是一种基于浏览器的软件架构模式，用户无需安装专用客户端，直接通过Web浏览器即可访问系统[9]。该架构将主要业务逻辑和数据处理集中在服务器端，客户端仅负责界面展示和简单交互，大幅降低了终端设备的性能要求。由于所有功能更新只需在服务端完成，系统维护和升级更加高效，特别适合企业级应用的大规模部署。随着云计算和Web技术的进步，B/S架构已突破早期网络带宽和服务器性能的限制，能够通过负载均衡、CDN加速等技术保障高并发场景下的稳定性，成为现代Web应用的主流选择[10]。

C/S（Client/Server）架构则采用客户端软件与服务器协同工作的模式，客户端需预先安装并承担部分计算任务，适用于对性能、实时性要求较高的场景，如大型游戏、工业控制软件等。该架构能充分利用本地硬件资源，支持复杂计算和离线操作，但同时也带来客户端兼容性、版本管理和部署成本等问题。在移动互联网时代，C/S架构仍广泛应用于需要深度设备交互的应用，如音视频处理、物联网终端等，但维护成本较高，需依赖自动更新机制或应用商店分发。

两种架构各有优势，B/S更适合轻量化、跨平台的Web服务，而C/S则适用于高性能、专业化的本地应用。随着WebAssembly、PWA等技术的发展，两者的界限正逐渐模糊，未来可能出现更多混合架构方案[11]。

## 2.2 前端框架Flutter和Vue

Flutter 是一款由 Google 开发的跨平台移动应用开发框架，使用 Dart 语言编写，支持一次开发、多端运行（iOS 和 Android），能够生成高性能的原生应用界面。Flutter 的组件化设计和热重载功能使得开发者可以快速迭代和调试代码，极大地提高了开发效率。Flutter 的核心优势在于其强大的跨平台能力，一套代码可以同时在 iOS 和 Android 平台上运行，大大降低了开发成本和维护工作量。它采用自绘引擎，不依赖于系统原生控件，这使得开发者能够完全掌控应用的 UI 设计和交互细节，实现高度定制化的用户界面。同时，Flutter 提供了丰富的 Material Design 和 Cupertino（iOS 风格）组件库，开发者可以根据项目需求快速搭建美观、易用的界面。

Vue 是一种渐进式的 JavaScript 框架，在本项目中主要用于构建 Web 端应用的用户界面。它以其简洁、灵活和易于上手的特性，成为前端开发团队的重要选择。其的主要目标是简化 Web 开发过程，提高开发效率，同时确保应用具备高性能和良好的可维护性。在开发模式上，Vue.js 采用声明式渲染，开发者只需描述应用的状态，即可自动渲染对应的视图，大大简化了 UI 开发过程。它还采用组件化开发模式，将复杂的 UI 拆分为多个独立、可复用的组件单元，每个组件都有自己的逻辑和样式，这不仅提高了代码的可维护性，还方便团队成员之间的协作开发。Vue.js 拥有庞大且活跃的社区，丰富的第三方资源，包括各类插件、工具和详细的文档，为开发者提供了强大的支持，无论是小型项目还是大型企业级应用，Vue.js 都能发挥其优势[13]。

## 2.3 后端框架Gin

Gin 是基于 Golang 开发的高性能、极简的 Web 框架，在本企业级项目中承担着后端服务的核心开发任务。Golang 作为一门由 Google 开发的开源编程语言，具有高效的并发性能、简洁的语法和丰富的标准库，为 Gin 框架的高性能和稳定性提供了坚实的基础。

Gin 框架的设计理念是轻量级和高效，它专注于提供简单而强大的路由功能、中间件支持以及高性能的 HTTP 处理能力。在路由方面，Gin 框架支持灵活的路由定义，能够根据不同的 URL 路径和 HTTP 方法将请求准确地分发到对应的处理函数。同时，它提供了丰富的中间件机制，开发者可以方便地添加诸如身份验证、日志记录、错误处理等通用功能，而无需在每个处理函数中重复编写代码。

Gin 框架充分利用了 Golang 的并发特性，通过轻量级的协程（goroutine）和通信机制（channel），能够高效地处理大量并发请求，非常适合开发高并发的企业级 Web 应用。在与数据库和其他后端服务的集成方面，Gin 框架具有良好的兼容性，能够方便地与 MySQL 等数据库进行交互，实现数据的存储、查询和更新等操作。

## 2.4 数据库MySQL

MySQL 是一款出色的关系型数据库管理系统，在本项目中负责存储和管理应用的核心数据。它以其强大的功能、高度的可扩展性和出色的稳定性，成为企业级应用数据存储的理想选择[13]。

MySQL 严格遵循 SQL 标准，并提供了丰富的高级特性，如复杂查询、外键约束、触发器、函数和视图等，能够满足企业级应用多样化的数据处理需求。它支持多种数据类型，包括 JSON、XML 等，使得存储和处理复杂结构的数据变得更加便捷。在事务处理方面，MySQL 拥有优秀的事务处理能力，通过 ACID（原子性、一致性、隔离性、持久性）特性确保数据的完整性和一致性，即使在高并发环境下也能保证数据的正确性。

在高并发场景下，MySQL 采用多版本并发控制（MVCC）机制，有效减少了读写操作之间的锁冲突，提高了系统的并发性能。此外，MySQL 拥有庞大的社区和丰富的扩展插件，开发者可以根据项目的具体需求，轻松扩展数据库的功能，如添加全文搜索、地理空间处理等功能。

## 2.6 分布式存储Redis

Redis（Remote Dictionary Server）是一种开源的分布式内存键值存储系统，其核心设计以内存数据结构服务器为基础，通过支持多种复杂数据结构（包括字符串、哈希、列表、集合、有序集合等）突破了传统键值存储系统的功能局限。作为高性能数据服务平台，Redis不仅实现了每秒数十万级读写操作的吞吐量，同时通过单线程事件驱动模型与非阻塞I/O机制的结合，在保障原子性操作的前提下有效避免了多线程并发控制带来的性能损耗，尤其在高并发场景下展现出显著的性能优势。该系统内置持久化机制，支持内存数据快照（RDB）与追加日志（AOF）两种磁盘存储策略，确保数据可靠性与服务容灾能力，并在此基础上构建了主从复制、哨兵监控和集群分片等特性，形成多层次的高可用架构。通过内嵌Lua脚本引擎，Redis支持将复杂操作封装为原子化事务，结合发布订阅模式、键空间通知及过期策略等扩展功能，使其在会话缓存、实时排行榜、分布式锁、消息队列等场景中展现出独特的工程价值。这种融合内存计算效率与磁盘持久化能力的双存储引擎设计，使其成为现代分布式系统中平衡性能与可靠性的典型解决方案[14]。

## 2.7 开发工具VS Code

VS Code（Visual Studio Code）是一款由微软开发的轻量级跨平台代码编辑器，在本项目的开发过程中发挥着重要作用。它支持多种编程语言，包括 Golang、Dart、JavaScript 等，能够满足项目中前后端开发的不同需求。

VS Code 具有强大的智能代码补全功能，能够根据代码上下文自动提示可能的代码选项，大大提高了代码编写的速度和准确性。同时，它提供了语法高亮、代码导航、代码格式化等功能，帮助开发者更清晰地阅读和理解代码结构，提高代码质量。其集成的调试工具功能全面，支持多种调试场景，包括单步调试、断点调试等，方便开发者快速定位和解决代码中的问题。

VS Code 的界面简洁直观，易于操作，支持多窗口和多标签页，开发者可以方便地同时处理多个项目。此外，VS Code 拥有丰富的插件生态系统，通过安装各种插件，开发者可以进一步扩展其功能，如支持特定的编程语言语法检查、代码分析、版本控制集成等。在团队协作方面，VS Code 与 Git 等版本控制系统无缝集成，方便团队成员进行代码管理和协作开发。

# **3** 系统分析与设计

## 3.1 需求分析

根据系统需求分析系统功能可详细划分为用户的登录注册，查看会议活动功能，搜索可用会议活动信息，提交预定申请，查看预定历史，预定成功通知，催办、用户管理、会议活动管理、预定管理、统计等[15]。

（1）用户的登录注册功能：在会议活动票务预定与签到管理系统中，用户的登录注册功能是非常重要的一部分。通过登录，用户可以使用系统中各种功能，包括查看自己的预定申请、查看预定历史等；而注册功能则允许新用户加入会议活动管理系统，享受到系统提供的服务。从功能需求的角度来看，用户的登录注册功能需要满足以下几点：一是用户注册，用户注册是指未曾在系统中登记过的用户通过填写表单等方式正式成为系统中的会员。在实现过程中，用户需要提供用户名、密码、手机号码等必要信息，同时可以填写其它选填信息。系统会根据输入的信息，验证用户名是否重复，验证手机号码是否正确和有效等。二是用户登录，用户登录是指在系统中已有账号的用户通过输入用户名和密码，验证身份后进入系统使用相应功能。

（2）会议活动浏览与选择：

会议活动票务预定与签到管理系统作为一个集成用户服务、和业务处理为一体的管理系统，用户应能够轻松浏览所有可用的会议活动，查看每个会议活动的详细信息，如位置、大小、可容纳人数、配备的设备等。提供直观的设计，确保用户在不同设备上（如手机、平板、电脑）都能良好地浏览和选择会议活动。

（3）会议活动预定与取消:

用户应能够在线查看会议活动的空闲状态，并根据需要预约合适的会议活动和时间段。提供在线预订和取消会议活动的选项，允许用户根据实际会议情况进行延时和结束操作。在预定过程中，系统应能够检测时间冲突，并提醒用户重新选择。

（4）会议通知与提醒：

系统应能够在会议预定成功、取消预订、会议变更时，通过电子邮件、短信、手机应用等方式向用户发送通知。提供会议前的自动提醒功能，确保与会人员不会遗忘或迟到。

（5）管理员管理会议活动和用户的功能：

管理员应该能够在系统中对于用户的突发状况进行管理，比如用户注销账号之后的删除，用户无法进入系统需要对于已经产生会议预定进行修改或者删除，帮助不能修改自己账户信息的用户进行信息修改。

1. 用户管理模块：

普通用户通过应用注册账号之后，就可以实现进行会议活动的预约行为，具体流程链表为：搜索活动 → 提交预定 → 订单支付 → 获取电子票 → 现场签到。

用户用例图如图3.1所示：

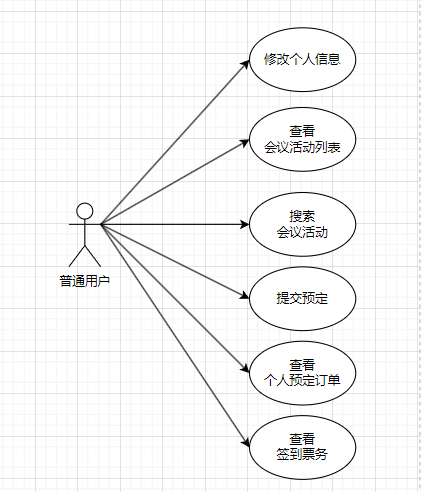


图3.1 用户用例图

（7）管理员模块：

管理员模块是会议活动票务预定与签到管理系统的一个关键功能模块，该角色的用户权限相对较高，作用是管理应用系统的数据，比如用户管理、会议活动管理、字典管理等等，其中字典管理用于统一系统中固定选项类数据（如性别、状态等）的标准值和显示文本，确保数据规范且便于维护。以确保系统的安全运行和信息保密性。管理员用例图如图3.2所示：

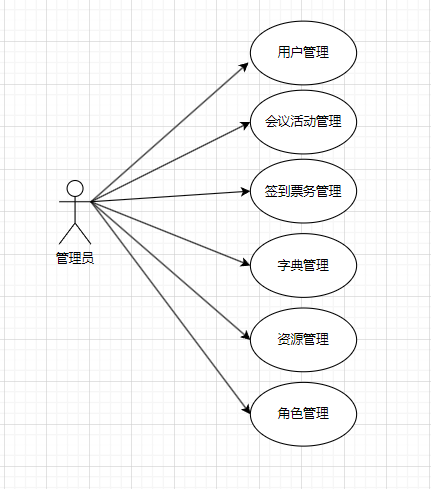


图3.2 管理员用例图

为了更加体现系统功能的完整性，绘制了系统功能模块图，如下图3.3 系统功能模块图。

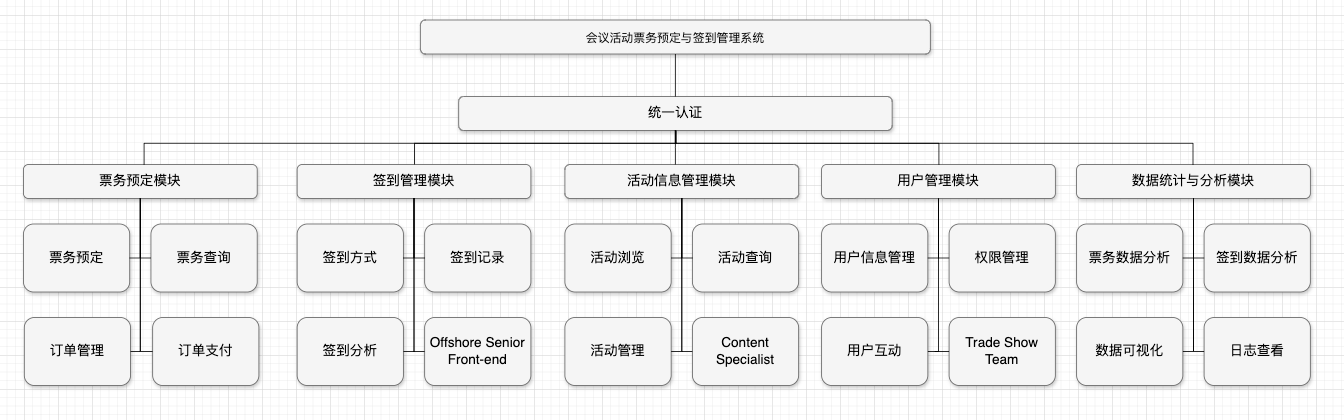


图3.3 系统功能模块图

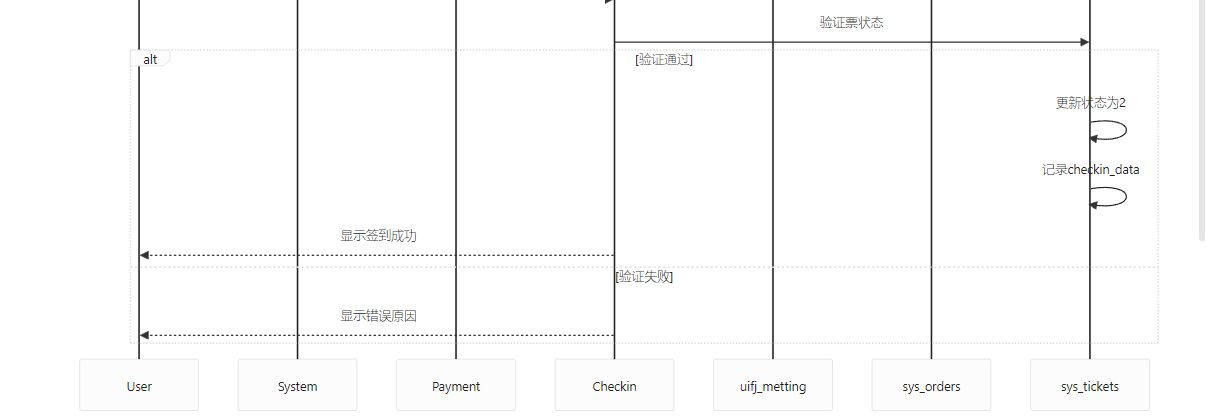
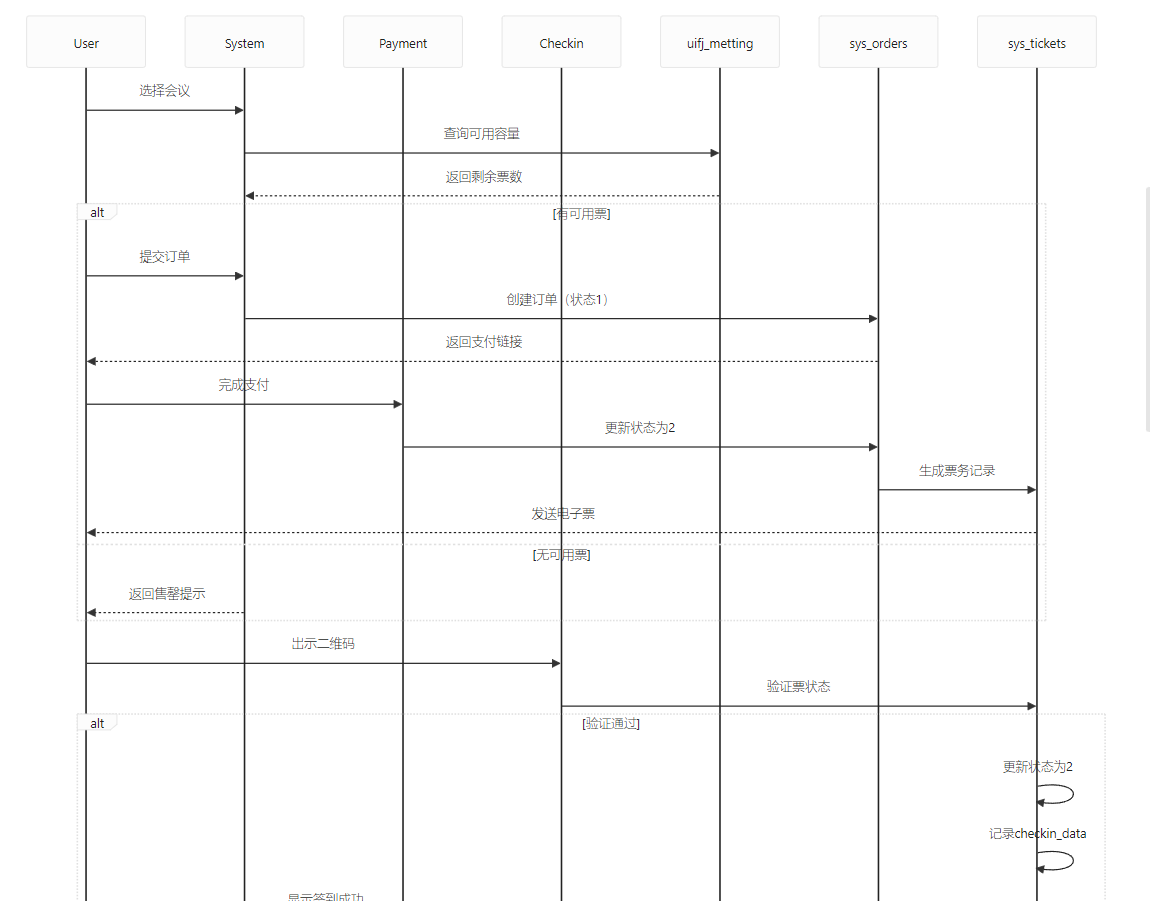
为了方便操作员以及会员更快熟悉系统的流程，编写了系统流程图，如下3.4 系统流程图：

图3.4 系统核心流程图

用例描述如表3.1、表3.2、表3.3所示：

表3.1 会议购票用例描述表

|  |  |
| --- | --- |
| **用例名称**： | 会议购票 |
| **参与者**： | 用户、会议活动系统 |
| **简要说明**： | 用户通过系统查询会议剩余票数并完成购票 |
| **前置条件：** | 用户已登录系统，会议信息已发布 |
| 基本事件流： | 1. 用户选择目标会议 2. 系统向票务模块查询可用容量 3. 票务模块返回剩余票数 4. 若有余票，系统创建待支付订单 5. 系统向用户返回支付链接 |
| 后置条件： | 生成待支付订单（状态1） |

表3.2票务支付用例描述表

|  |  |
| --- | --- |
| **用例名称：** | **用户、支付系统** |
| **参与者：** | **用户完成订单支付并获取电子票** |
| **简要说明：** | **用户完成订单支付并获取电子票** |
| **前置条件：** | **存在未支付订单（状态1）** |
| **基本事件流：** | 1. **用户通过支付链接完成付款** 2. **支付系统通知订单系统更新状态** 3. **订单系统将订单标记为已支付（状态2）** 4. **票务系统生成电子票并发送至用户** |
| **后置条件：** | **订单状态变更为已支付，电子票存入用户账户** |

表3.3会议签到用例描述表

|  |  |
| --- | --- |
| **用例名称：** | **会议签到** |
| **参与者：** | **用户、签到设备** |
| **简要说明：** | **用户通过二维码完成会议入场核验** |
| **前置条件：** | **用户持有有效电子票（状态2）** |
| **基本事件流：** | 1. **用户向签到设备出示电子票二维码** 2. **设备验证票务状态有效性** 3. **若验证通过，更新票务状态为已使用（状态3）** 4. **记录签到时间及设备信息** 5. **显示签到成功提示** |
| **后置条件：** | **票务状态变更为已使用，生成签到记录** |

其中部分功能描述如下：

(1)用户登录注册：用户输入自己的用户名和密码登录该系统，如果登录成功即可显示系统主界面。如果显示账号或密码不正确则登录失败，则需要用户进行注册或联系管理员重置密码。

(2)个人信息管理：成功登录到系统的用户可以进行个人信息管理，进入个人中心可以查看到自己的具体信息，并且可以对自己的个人信息进行修改。

(3)管理员：管理员可以对其会议信息进行查看，能够查看会议活动信息、可预约人数、订单号、介绍详情等信息。

(4)订单生成：用户预定后，弹出订单页面。订单号生成规则：`DATE\_FORMAT(NOW(), '%Y%m%d') + SUBSTR(UUID(), 1, 8)`，库存冻结机制：创建订单时预扣减剩余容量（需配合事务处理）。

(5)票务签到：用户支付订单后，自动生成对应会议活动的票务证据。票号生成规则：`SHA256(CONCAT(meeting\_id, user\_id, UNIX\_TIMESTAMP()))`，二维码数据加密：采用AES-256加密存储参会人基本信息。

订单状态流程图如图3.5所示：

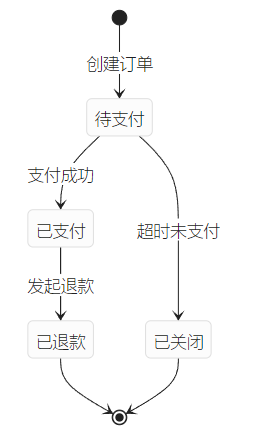


图3.5 订单状态流程图

订单状态流程图如图3.6所示。

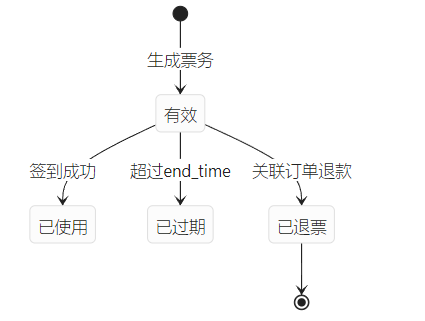


图3.6 订单状态流程图

管理员通过扫描用户票务二维码核销用户会议活动票务数据。具体过程图如图3.7所示：



图3.7 签到管理过程图

## 3.2 系统可行性分析

3.2.1 经济可行性

本系统采用开源技术栈构建核心模块，相较于商业软件授权模式可降低80%的初期投入成本。云基础设施层可以依托阿里云ACK容器服务或者通过腾讯云计算实现按需付费，结合Serverless架构设计使资源利用率提升40%以上，综合测算可使年度运维成本下降35%。业务流程重构方面，自动化票务管理系统预计减少人工核销岗位60%的工作量，电子票务替代方案每年可为大型场馆节约纸质耗材费用超百万元。收益增长维度分析表明，基于用户行为数据的动态定价模型可使座位利用率提高15个百分点，结合精准营销策略预计实现12%-18%的收入增长。项目投资预算涵盖6人月的全栈开发成本与弹性伸缩的云资源支出，经财务模型测算静态回收期为8.7个月，动态投资回报率可达行业基准值的1.3倍，成本效益比处于同业领先水平[16]。

3.2.2 技术可行性

经济可行性评估显示，开源技术栈节省80%商业软件许可费用，云服务按需付费模式降低35%运维成本。系统带来的自动化管理可减少50%人工成本，电子票每年节省百万级印刷费用，数据驱动的定价策略能提升12-18%收入。投资回报周期约9个月，成本效益比达到行业领先水平。系统建设总成本包含6人月的开发投入和弹性可调的云资源支出，经济效益主要体现在运营成本节约和收入增长两个维度。

3.2.3 操作可行性

操作可行性层面，系统严格遵循人机交互国际标准，购票流程三步完成使操作效率提升62%，移动端支持主流电子钱包并符合无障碍标准。管理后台提供可视化库存面板和智能预警功能，管理员仅需4小时培训即可掌握核心操作，学习成本低于同类产品27%。双端设计均通过用户测试验证，其中93%测试者能在15秒内完成票务调取，操作便捷性得到充分保障[17]。

3.2.4 时间可行性

时间可行性采用敏捷开发分三期实施，3个月完成核心功能开发，2个月实现增强模块，最后1个月进行系统优化和安全审计。项目设置20%风险储备时间，关键路径无强依赖冲突，基于历史数据测算按期交付概率达92%。同类技术架构项目平均延期率较行业低15个百分点，实施进度可控性强。

## 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：

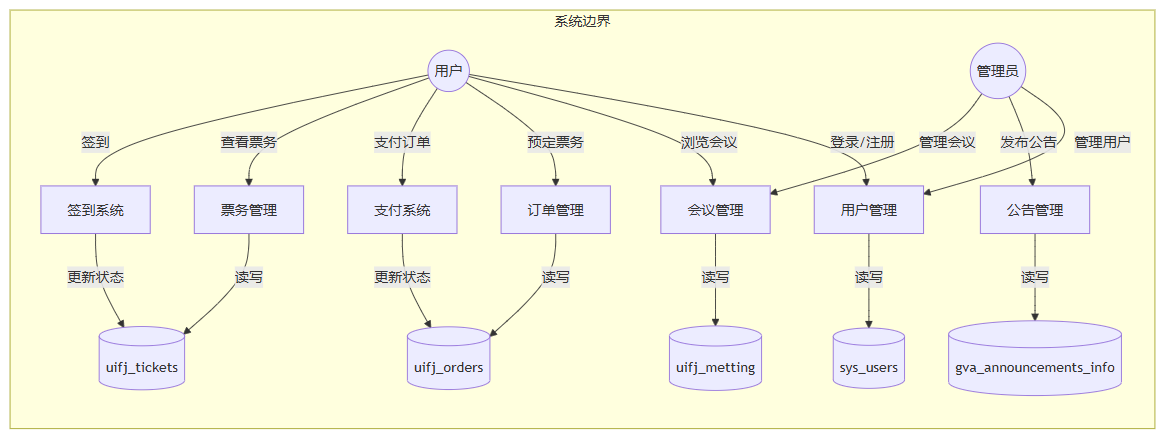


图3.8 系统数据流图

## 

## 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：

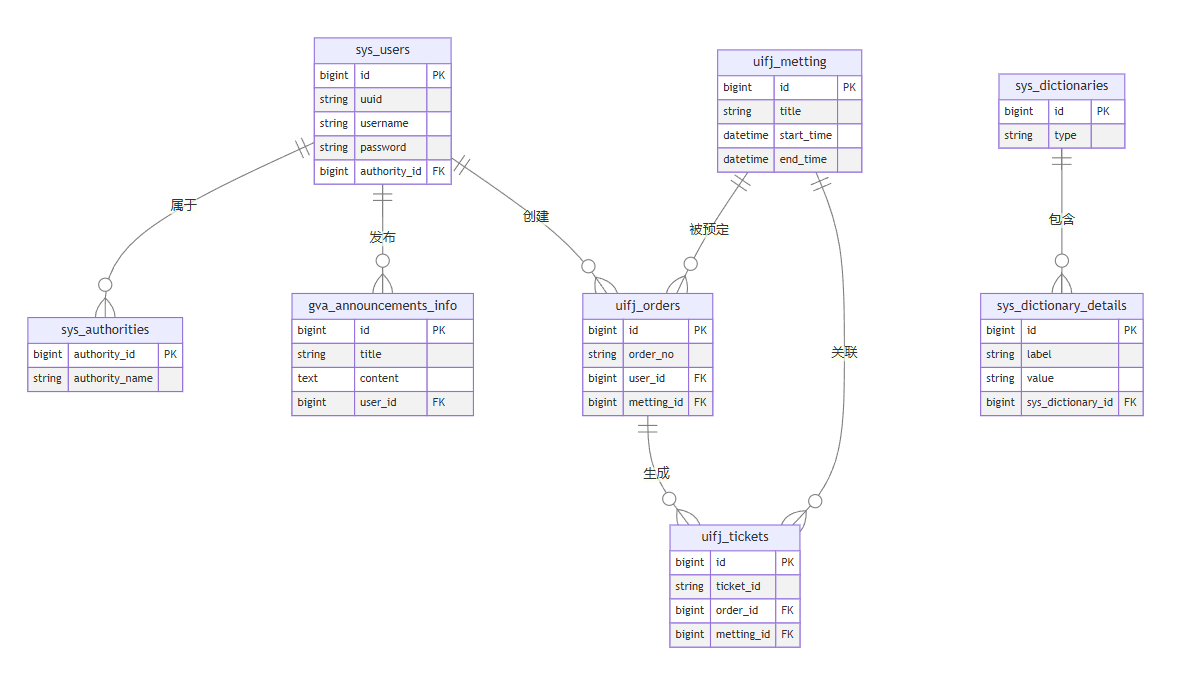


图3.9 系统E-R图

用在会议活动管理系统中，MySQL数据库的设计是非常重要的。数据库设计要充分考虑系统的需求和业务逻辑，合理地设计表结构和关系，以保证数据的完整性和一致性。

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3.9所示：

表3.4 casbin\_rule（权限规则表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| ptype | varchar | 100 | YES |  | 策略类型（p-策略，g-角色） |
| v0 | varchar | 100 | YES |  | 主体标识（角色/用户） |
| v1 | varchar | 100 | YES |  | 资源路径 |
| v2 | varchar | 100 | YES |  | 操作方法 |
| v3 | varchar | 100 | YES |  | 策略扩展字段1 |
| v4 | varchar | 100 | YES |  | 策略扩展字段2 |
| v5 | varchar | 100 | YES |  | 策略扩展字段3 |

表3.5 sys\_authorities（角色表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| authority\_id | bigint |  | NO |  | 角色ID（自增主键） |
| authority\_name | varchar | 191 | YES |  | 角色名称 |
| parent\_id | bigint |  | YES |  | 父级角色ID |
| default\_router | varchar | 191 | YES | dashboard | 默认路由 |

表3.6 sys\_dictionaries（字典主表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| name | varchar | 191 | YES |  | 字典名称（中文） |
| type | varchar | 191 | YES |  | 字典类型（英文标识） |
| status | tinyint |  | YES |  | 状态（0-启用 1-禁用） |
| desc | varchar | 191 | YES |  | 字典描述 |

表3.7 sys\_dictionary\_details（字典明细表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| label | varchar | 191 | YES |  | 显示标签 |
| value | varchar | 191 | YES |  | 实际值 |
| extend | varchar | 191 | YES |  | 扩展字段 |
| status | tinyint |  | YES |  | 状态（0-启用 1-禁用） |
| sort | bigint |  | YES |  | 排序序号 |
| sys\_dictionary\_id | bigint |  | YES |  | 关联字典主表ID |

表3.8 gva\_announcements\_info（公告信息表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 主键 |
| title | varchar | 191 | YES |  | 公告标题 |
| content | text |  | YES |  | 公告内容 |
| user\_id | bigint |  | YES |  | 发布者ID |
| attachments | json |  | YES |  | 附件信息（存储路径数组） |

表3.9 sys\_users（系统用户表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 用户唯一标识 |
| uuid | varchar | 191 | YES |  | UUID身份凭证 |
| username | varchar | 191 | NO |  | 登录账号（唯一） |
| password | varchar | 191 | NO |  | BCrypt加密密码 |
| nick\_name | varchar | 191 | YES | 系统用户 | 显示名称 |
| header\_img | varchar | 191 | YES | 默认头像URL | 头像地址 |
| authority\_id | bigint |  | YES | 888 | 关联角色ID |
| phone | varchar | 191 | YES |  | 11位手机号 |
| email | varchar | 191 | YES |  | 标准邮箱格式 |
| enable | bigint |  | YES | 1 | 账户状态（1启用/2冻结） |

表3.10 uifj\_metting（会议主表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 会议唯一ID |
| title | varchar | 191 | NO |  | 会议名称（50字内） |
| capacity | bigint |  | NO |  | 最大参与人数（≥1） |
| location | varchar | 191 | YES |  | 经纬度坐标（WGS84） |
| type | varchar | 191 | NO |  | 类型（1线下/2线上） |
| status | varchar | 191 | NO |  | 状态（1未开始/2进行中/3已结束） |
| posturl | varchar | 191 | YES |  | 封面图URL地址 |
| equipment | varchar | 191 | YES |  | 设备清单（逗号分隔） |
| description | varchar | 191 | YES |  | 会议简介（200字内） |
| start\_time | datetime | 3 | NO |  | 起始时间 |
| end\_time | datetime | 3 | NO |  | 结束时间 |

表3.11 uifj\_orders（订单表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 订单主键 |
| order\_no | varchar | 24 | NO |  | 订单号 |
| user\_id | bigint |  | NO |  | 关联用户表ID |
| metting\_id | bigint |  | NO |  | 关联会议表ID |
| ticket\_type | tinyint |  | NO | 1 | 票种（1普通票/2VIP票） |
| quantity | int |  | NO |  | 购买数量（1-999） |
| unit\_price | decimal | 10,2 | NO |  | 单价（两位小数） |
| total\_price | decimal | 10,2 | NO |  | 总金额（两位小数） |
| pay\_status | tinyint |  | NO | 1 | 支付状态（1待付/2已付/3关闭/4退款） |
| pay\_time | datetime | 3 | YES |  | 支付成功时间（精确到毫秒） |
| refund\_no | varchar | 32 | YES |  | 退款单号（字母+数字组合） |

表3.12 uifj\_tickets（票务信息表）

| 字段名 | 类型 | 长度 | 是否为空 | 默认值 | 注释 |
| --- | --- | --- | --- | --- | --- |
| id | bigint |  | NO |  | 票据唯一ID |
| ticket\_id | varchar | 32 | NO |  | 票据编号（SHA256哈希值） |
| order\_id | bigint |  | NO |  | 关联订单表ID |
| user\_id | bigint |  | NO |  | 关联用户表ID |
| metting\_id | bigint |  | NO |  | 关联会议表ID |
| status | tinyint |  | NO | 1 | 状态（1有效/2已用/3过期/4退票） |
| qr\_data | varchar | 191 | NO |  | AES加密的二维码数据 |
| checkin\_data | json |  | YES |  | 签到信息（设备ID+时间+位置坐标） |

# **4** 系统实现

## 4.1 系统目录结构

会议活动票务与签到管理系统分用户、管理员、超级管理员三个主体。用户借助前端界面来与系统实现交互，管理员、超级管理员则通过后端界面来与系统进行连接，从而实现对会议活动票务与签到管理系统各个功能模块的管理。系统目录图如4.1图所示：

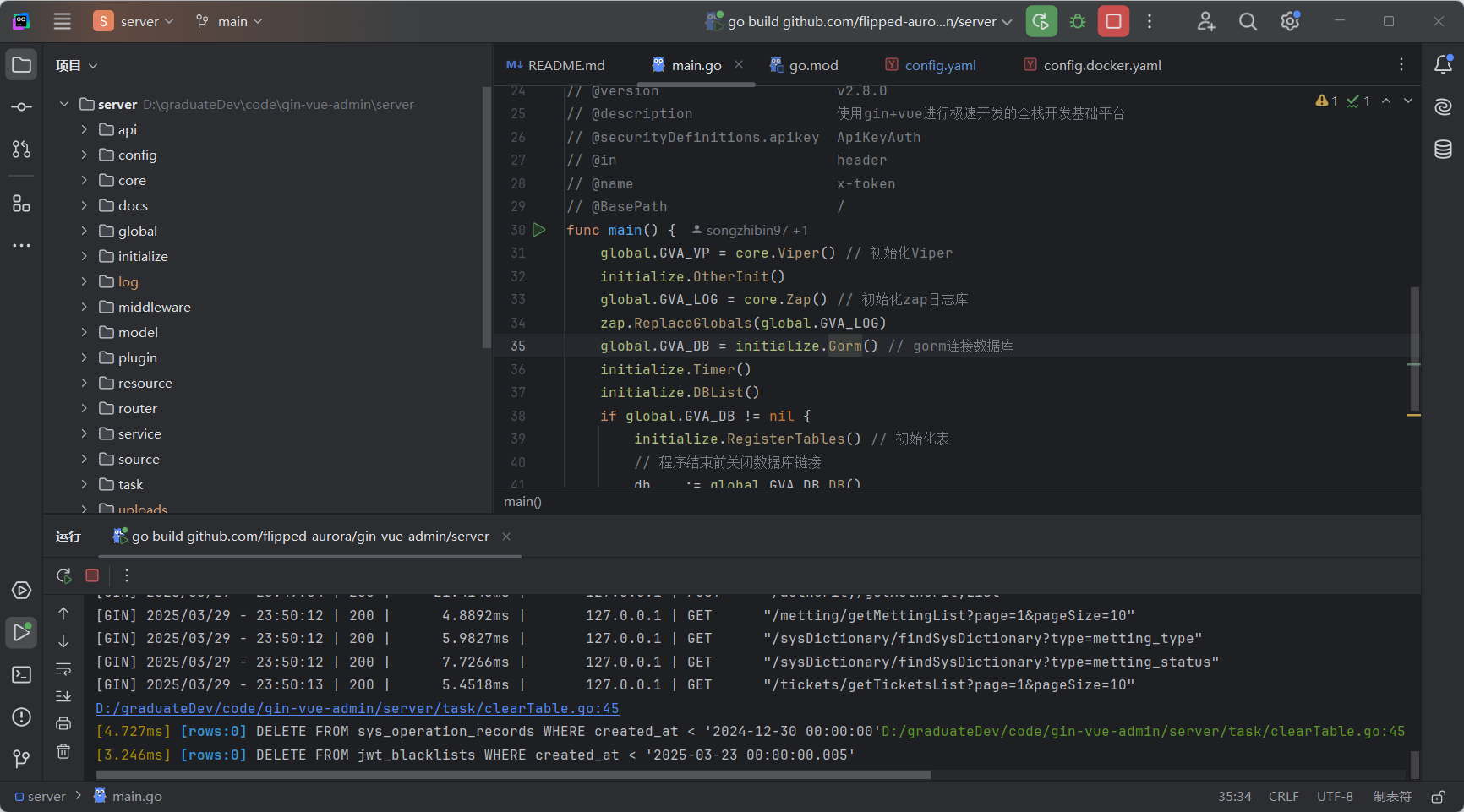


图4.1 系统目录图

## 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:8081/#/layout/dashboard进入会议活动票务与签到管理系统的管理首页，用户可以点击“登录”或“注册”进入自己账号的系统首页。首页的设计布局方式基于Element的组件系统做出来的，进入系统前端首页，可以查看到仪表盘、会议活动信息、公告信息、个人中心等功能模块，并可以进行详细的操作。界面如图4.2所示：



图4.2 系统前端首页

4.2.2 注册登录模块

当用户需要预定会议活动时，需要先登录本系统前端应用模块方可进行预定操作。如果用户尚未注册，必须先完成注册流程。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、密码等在内的个人信息。一旦注册成功，用户便能够通过输入相应的账号和密码登录系统，并顺利进入首页，从而继续进行预定操作。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：

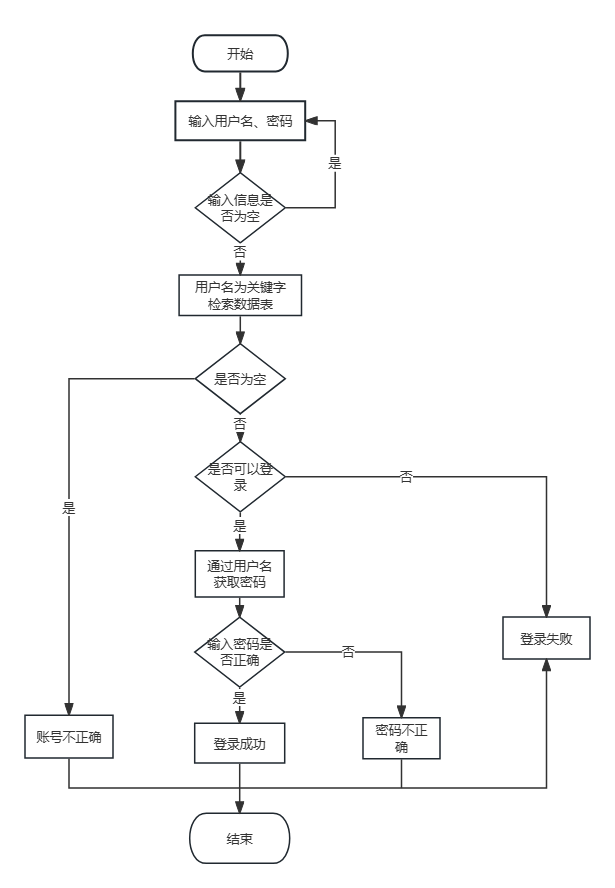


图4.3 用户登录流程图



图4.4 用户注册界面



图4.5 用户登录界面

4.2.3 会议活动信息模块

用户进行登录操作后，在前台首页的会议活动信息模块可以查看到系统所有的会议活动列表，可以查看会议活动信息、举办时间和地点等。在该模块展示的所有会议活动列表里，可以点击查看到会议活动的具体详细信息，会议活动类型，可预约人数，价格详情等信息，可以对该会议活动进行预定操作。点击“预定票”按钮即可实现票务预定效果。如图4.6、图4.7、图4.8所示：

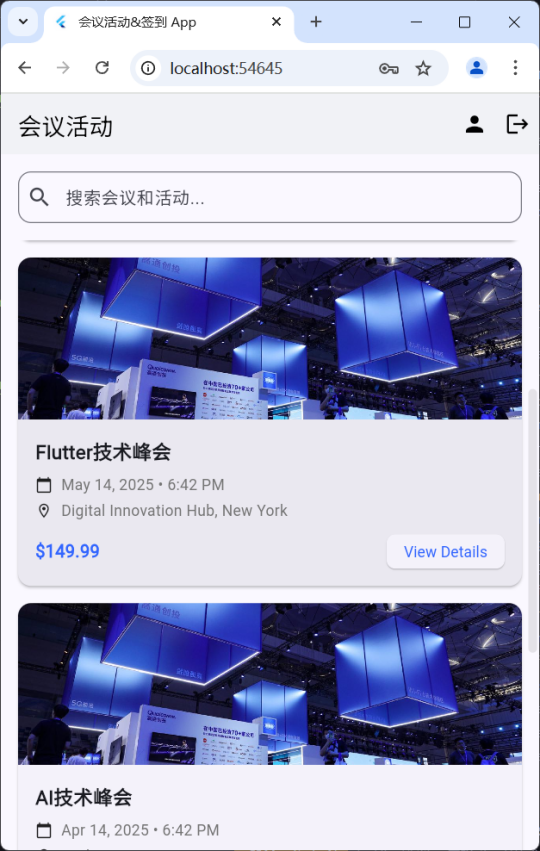


图4.6 会议活动信息展示界面

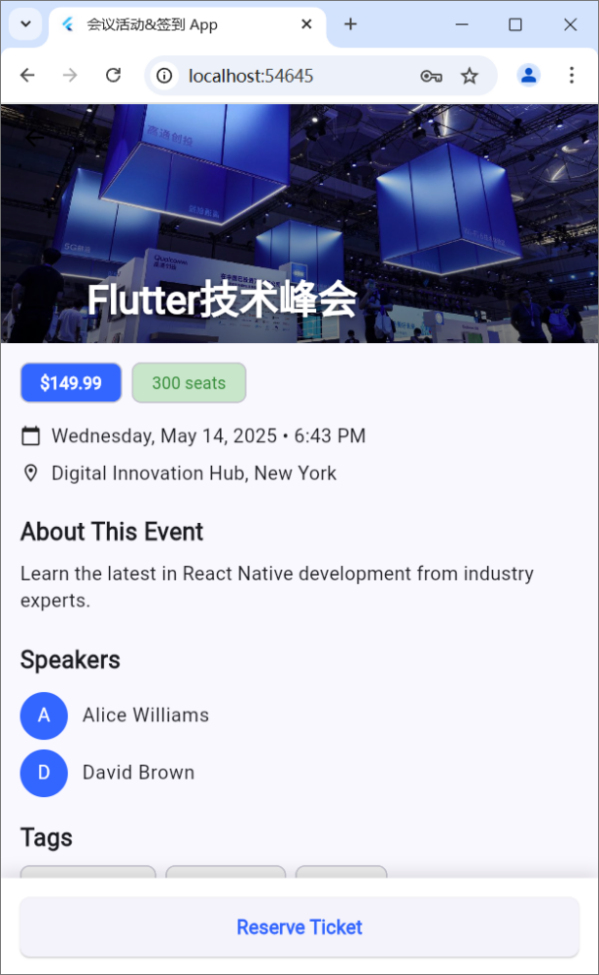


图4.7 会议活动详情

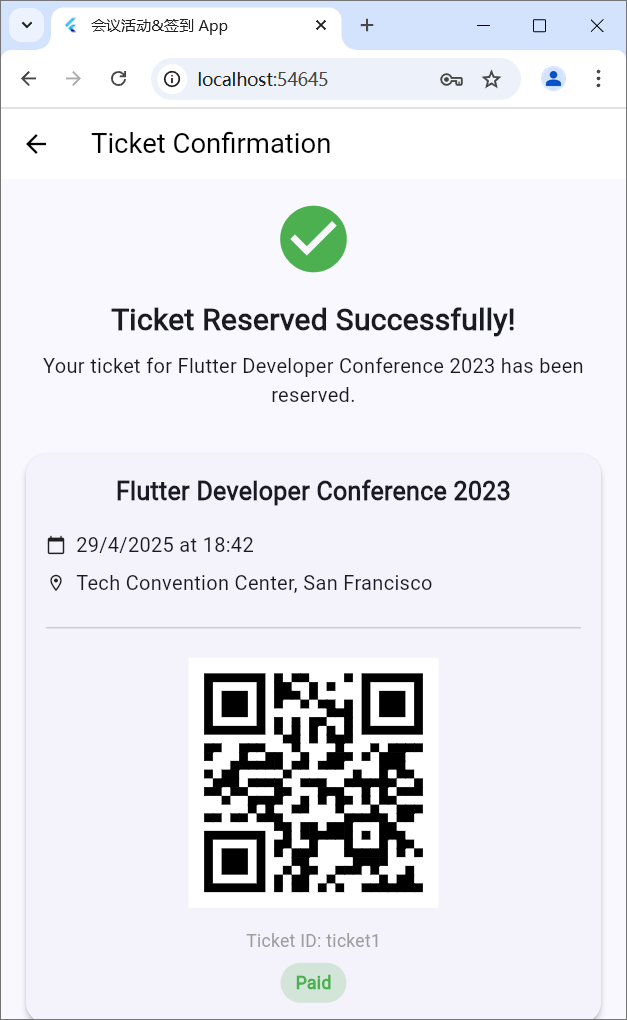


图4.8 会议活动票务预约下单界面

4.2.4 订单票务模块

用户在前台首页的订单票务模块可以查看到系统所有的订单详情和支付详细信息，可以通过票务信息来对当前预约成功的票务进行单一的查询操作，用户在支付成功后，会弹出具体会议活动的票务信息。工作人员可以通过工作员账号，使用扫描功能核销用户预定的会议活动。具体如图4.9、图4.10、4.11所示：

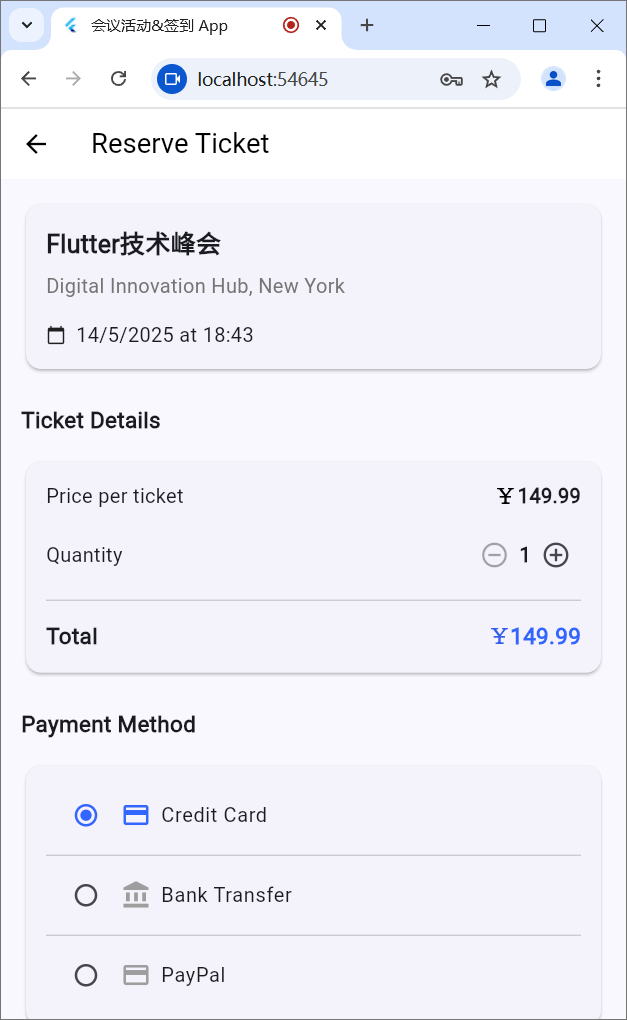


图4.9 支付详情界面

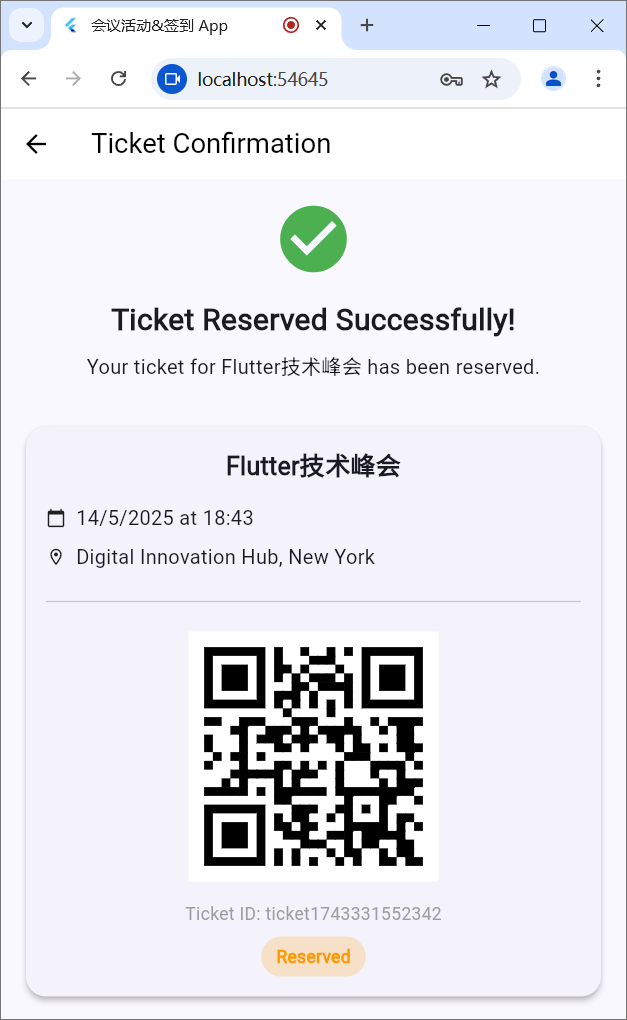


图4.10 票务预定成功页面

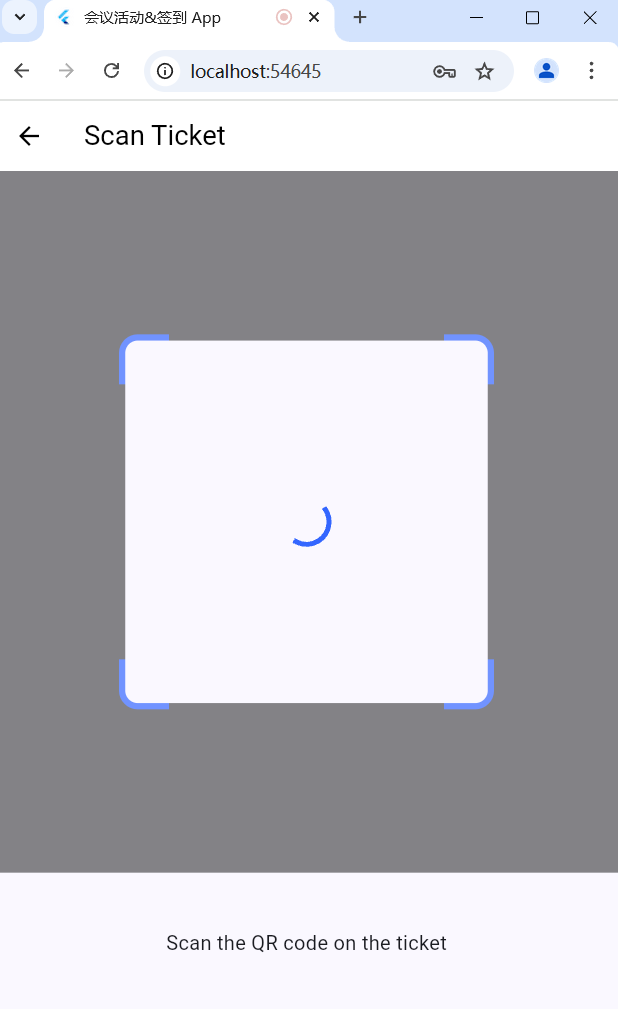


图4.11 扫票检票界面

4.2.5 个人中心模块

用户在个人中心模块可以查看到个人详细信息，会议活动订单等模块，并对该些功能模块进行详细的操作，在个人信息模块可以对头像和性别等信息进行更新。可以选择退出登录操作。在订单信息详情可以查看到所有会议活动订单以及已支付订单，已退款订单和已评价订单。具体如图4.12所示：

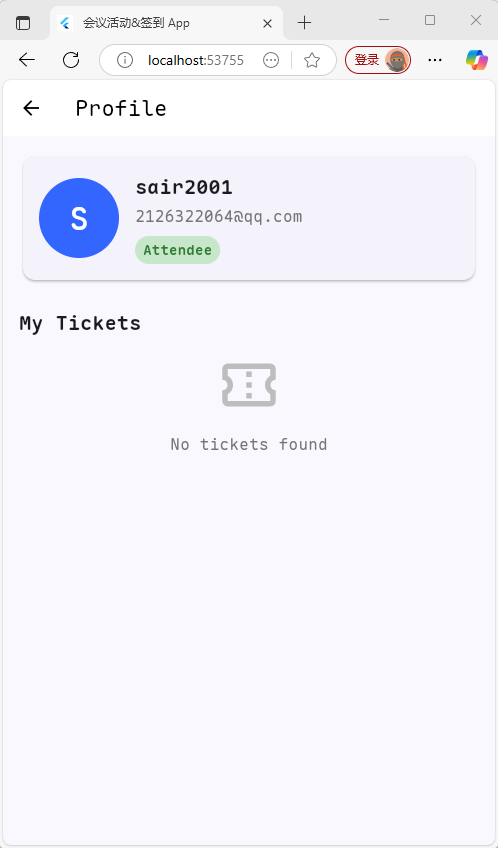


图4.12 个人中心详情

## 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4.13所示，相关界面展示如图4.14所示：

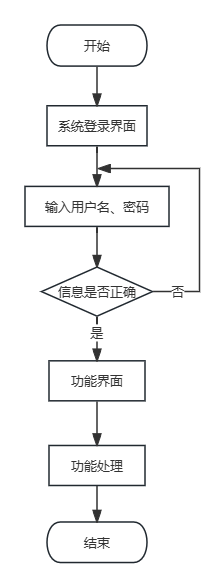


图4.13 管理员、管理员登录流程图



图4.14 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，可以对用户的昵称、手机号、邮箱等信息进行修改。如图4.15、图4.16所示：



图4.15 用户管理



图4.16 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角色进行管理，在该模块也能新增角色到系统中来，可以为不同的角色授予权限，根据不同角色创建用户，对系统具有不同的操作权限。如下图4.17所示：



图4.17 角色管理

超级管理员在角色管理模块可以对系统所有的角色进行分配权限，不同的用户具有不同权限，权限功能能让该用户所属的角色访问相应的接口和数据。如下图4.18所示：



图4.18 角色权限设置

4.3.4 字典管理模块

管理员在字典管理模块可以对系统所有字典信息进行修改，删除等操作。字典管理用于统一系统中固定选项类数据（如性别、状态等）的标准值和显示文本，确保数据规范且便于维护。界面如图4.19所示：

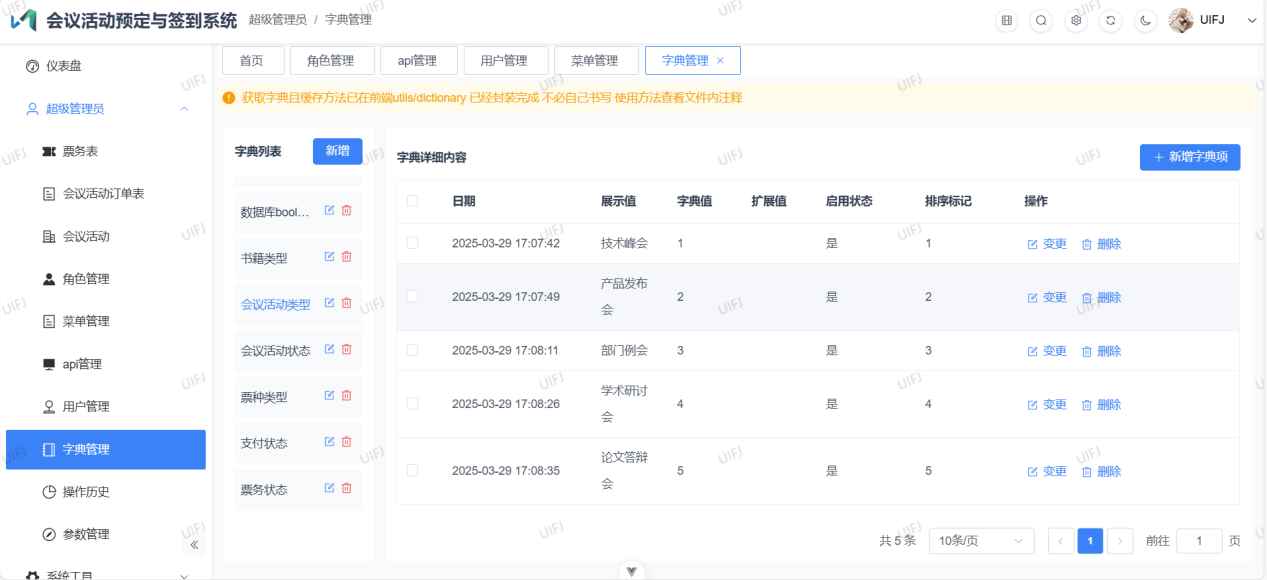


图4.19 字典管理

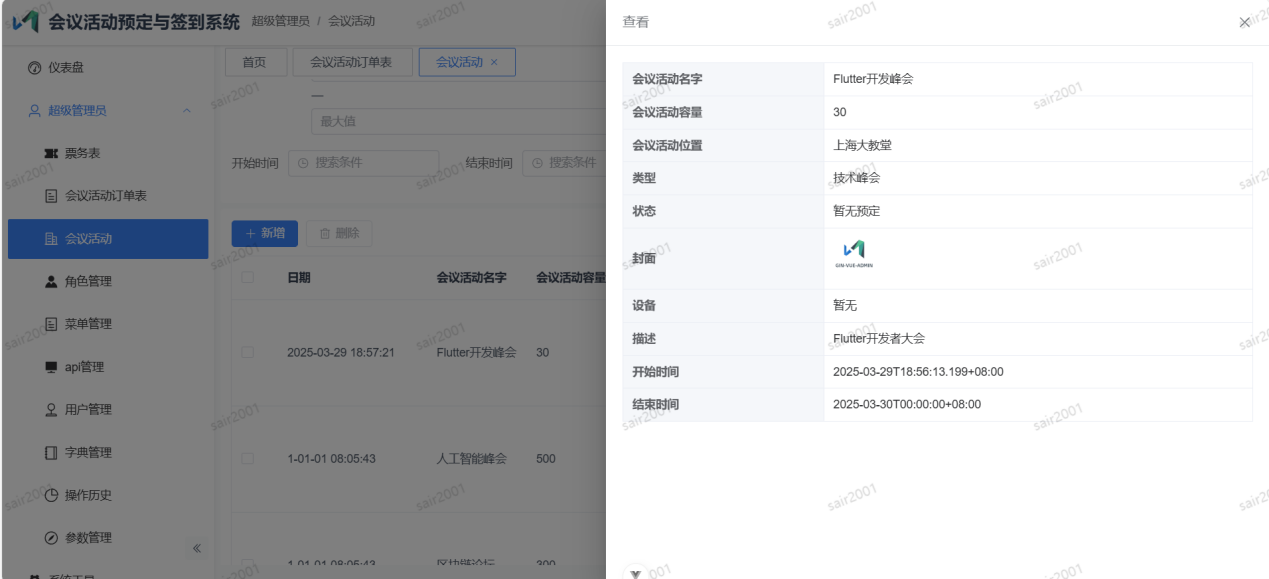
4.3.5 会议活动信息管理模块

管理员在会议活动信息管理列表可以查看到系统所有的会议活动详情，主要包括会议活动的名称、宣传报、会议活动地点、类型、状态、会议活动类型、预定价格等会议活动信息，可以对对应信息进行修改、删除、新增和查询的操作，在该模块可以对会议活动可预约人数进行减少和增加的修改。界面如图4.20所示：



图4.20 会议活动信息管理

管理员在该模块可以查看到会议活动具体信息。具体实现界面如图4.21所示：

图4.21 管理员会议活动查看

管理员在会议活动信息管理中能查看管理会议活动信息，主要包括会议活动名称、宣传图、开始时间、结束时间、会议活动类型、可预约人数和价格等会议活动信息。界面如图4.22所示：



图4.22 管理员修改会议活动信息

4.3.6 订单管理模块

管理员在商品管理模块可以查看到系统展示在前台的所有会议活动详情主要包括用户ID、会议活动ID、订单号等信息，可以对该信息进行修改、删除、新增的操作，在该界面可以对订单进行退款操作。

管理员在该模块可以查看到用户在前台的支付记录和订单具体情况，然后进行对应的操作。具体实现界面如图4.23、4.24、4.25所示：



图4.23 订单管理

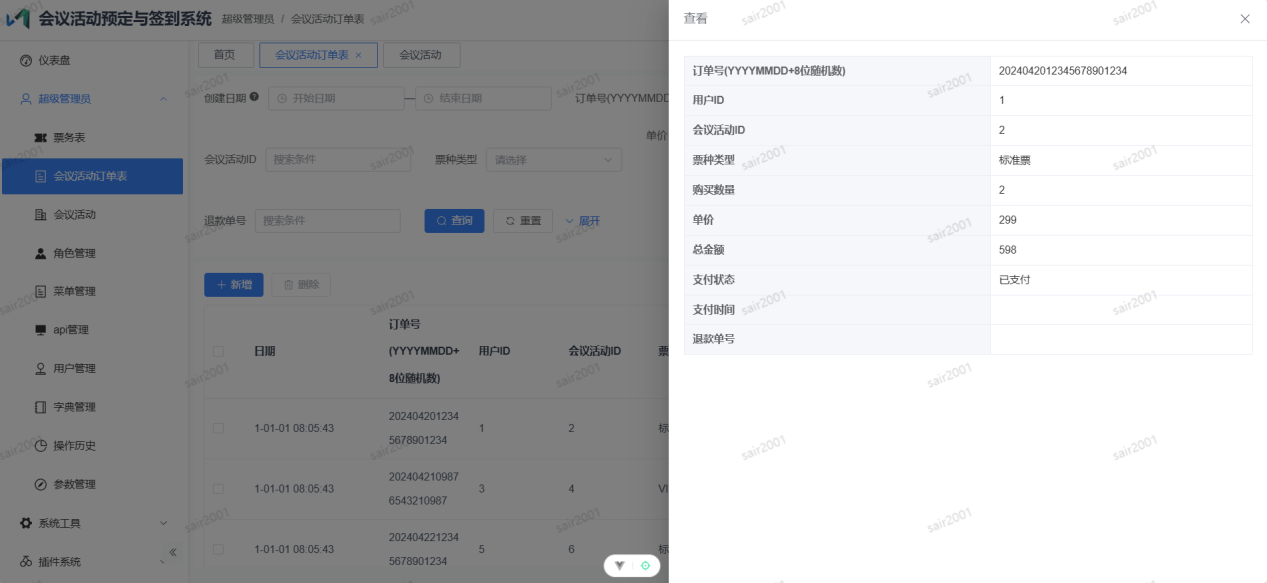


图4.24 订单详情查看

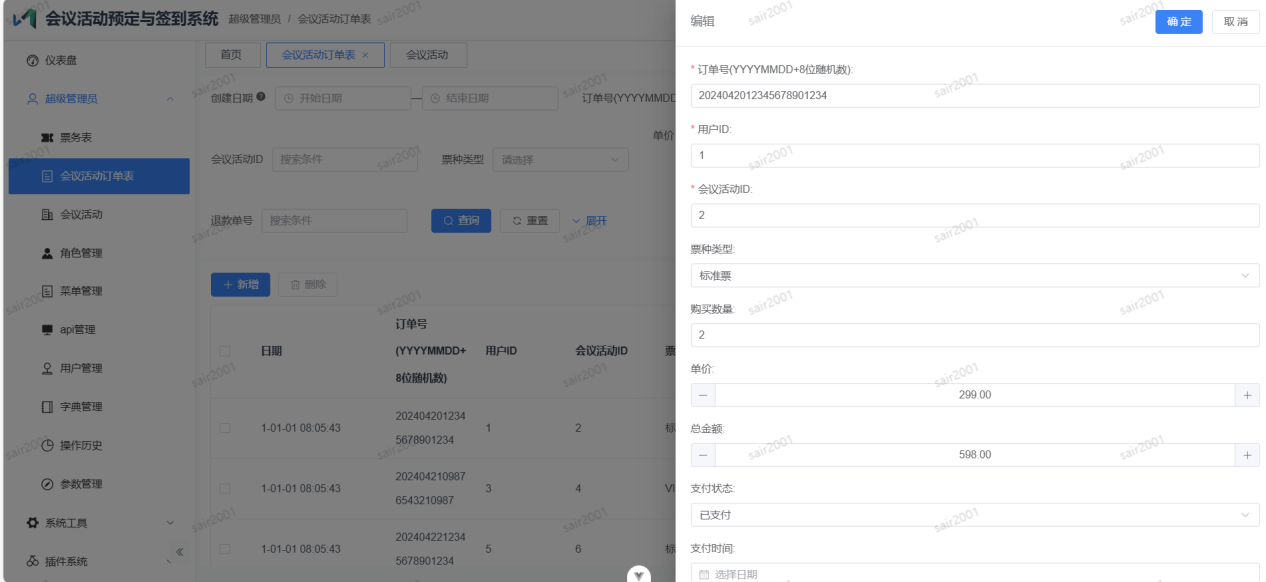


图4.25 订单信息编辑

4.3.7 票务管理模块

管理员在基础数据管理模块可以管理系统展示用户的所有预定的会议活动后出示的票务信息。让签到管理的运营更方便，具体实现界面如图4.26、4.27、4.28所示：

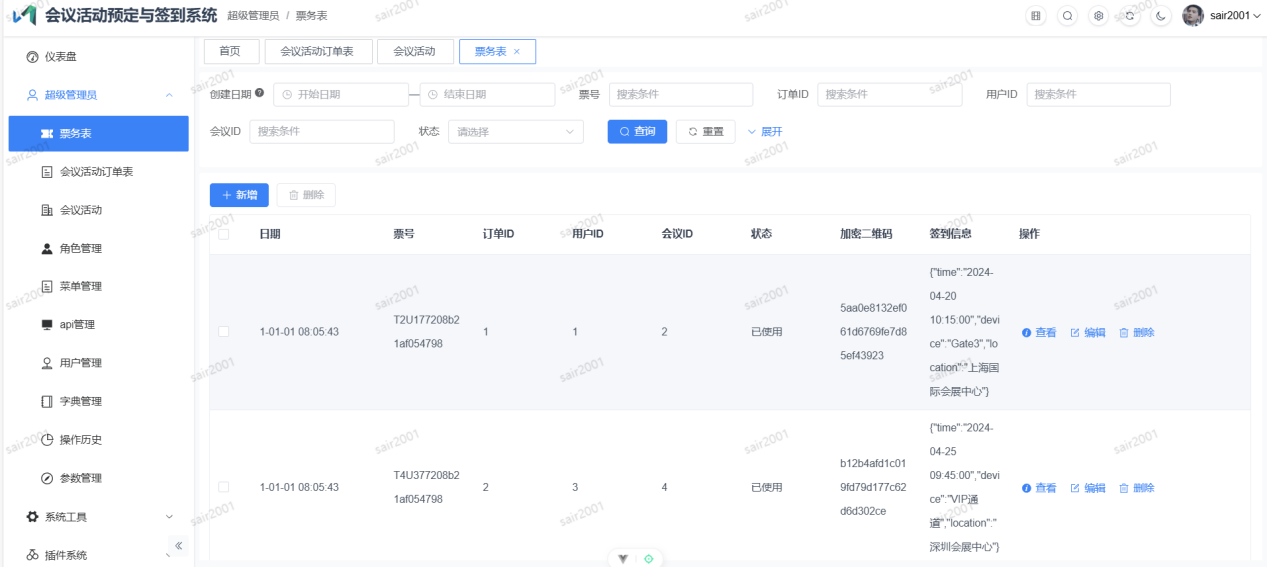


图4.26票务签到管理

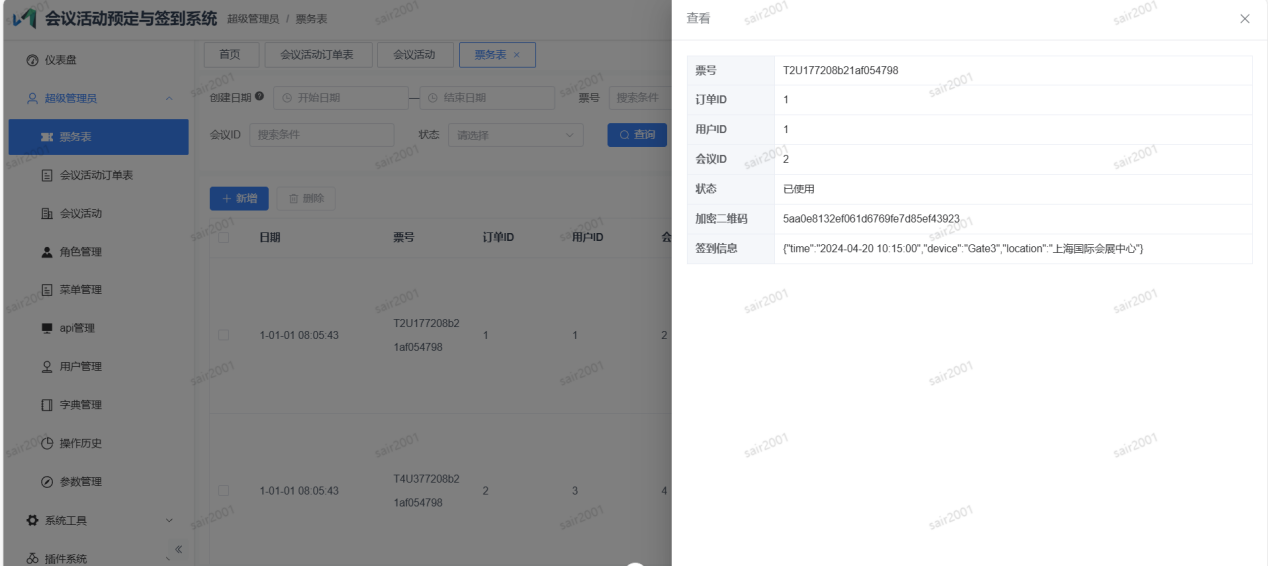


图4.27 票务查看

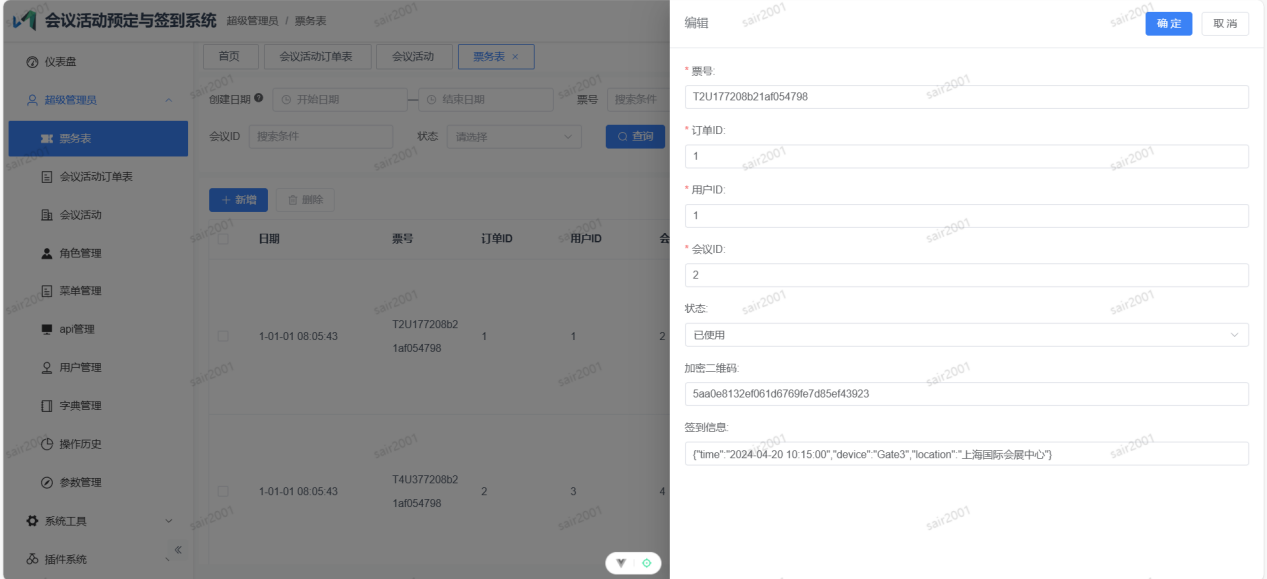


图4.28 票务编辑

4.3.8 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、4.30所示：



图4.29 管理员公告信息管理

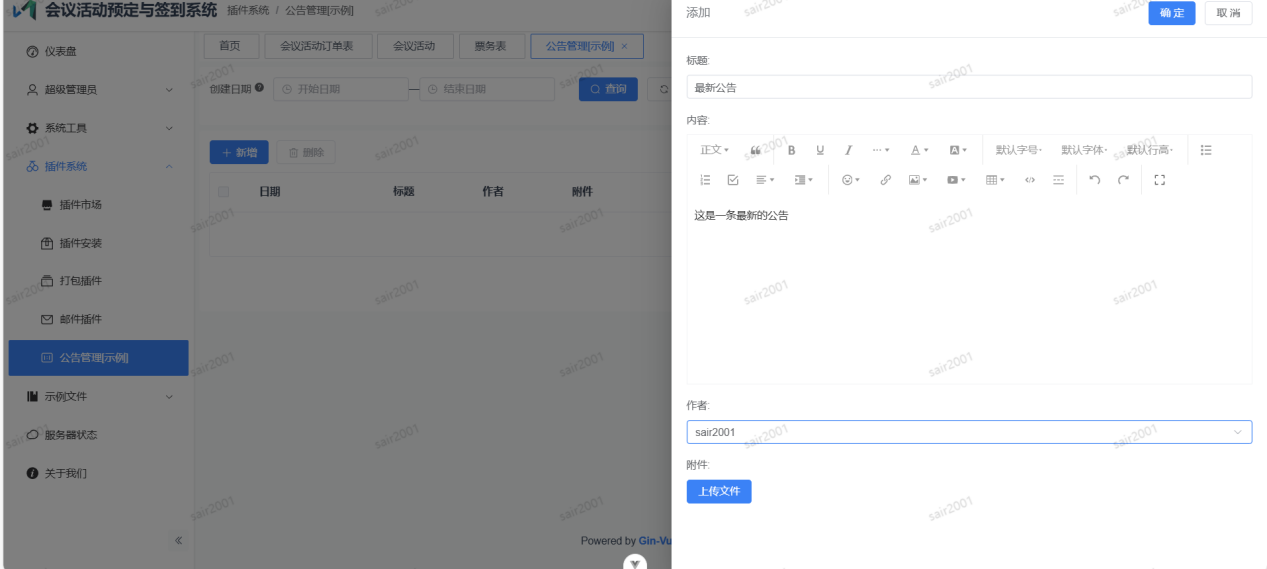
管理员在公告信息模块可以查看系统展示在前台界面的公告详情，界面如图4.30所示： 

图4.30 管理员公告信息发布

# 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 数据库初始化功能测试

系统在开始部署运行时，点击“初始化”会出现两种情况：初始化成功跳转登录界面或初始化失败，弹出提示。初始化功能测试用例表如表5.1所示，初始化功能测试结果图如图5.1所示。

表5.1数据库初始化功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 操作序号 | 操作描述 | 预期输出 | 实际输出 | 结果 |
| 1 | 未通过docker连接数据库初始化 | 没有数据库连接 | 与预期结果相同 | 通过 |
| 2 | 初始化数据库密码错误 | 数据库密码错误 | 与预期结果相同 | 通过 |
| 3 | 已经初始化了数据库 | 已配置数据库信息，无法初始化 | 与预期结果相同 | 通过 |



图5.1　数据库初始化功能测试结果图

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：登录成功跳转会议活动票务与签到管理系统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，查看自己的会议活动订单和商品兑换记录，对购买的会议活动进行评价或退款。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 | 进入会议活动订单，选择会议活动点击评价，输入评价并提交 | 返回会议活动订单页面，订单类型由已支付更新为已评价 | 与预期结果相同 | 通过 |
| 3 | 进入会议活动订单，选择会议活动点击退款 | 弹出选项框“确认退款或取消”，确认后返回会议活动订单页面，订单类型由已支付更新为已退款 | 与预期结果相同 | 通过 |

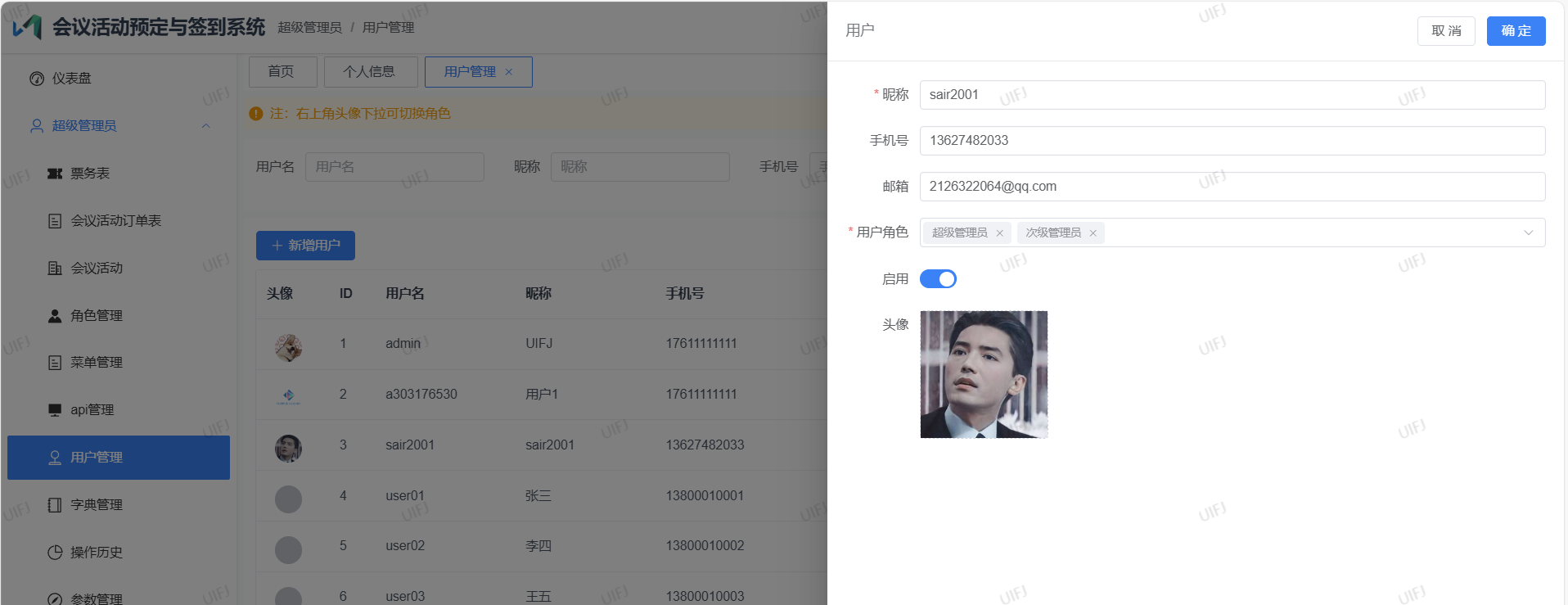


图5.3　个人中心功能测试结果图

5.2.4 会议活动信息管理功能测试

在会议活动信息管理界面中，管理员能够看到系统中所有会议活动的信息以一个列表的形式展现出来，并且还可以对会议活动进行信息的修改，会议活动信息管理测试用例表如表5.4所示，会议活动信息管理测试结果图如图5.4所示。

表5.4 会议活动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 | 在管理员页面点击会议活动信息管理 | 跳转到会议活动信息管理页面，以分页列表形式显示所有会议活动信息 | 跳转到会议活动信息管理页面，以分页列表形式显示所有会议活动信息 | 通过 |
| 2 | 在会议活动信息管理页面点击新增，在添加会议活动信息页面输入会议活动信息信息后点击提交 | 成功添加会议活动信息，跳转到会议活动信息管理页面并在会议活动信息列表中显示新增的会议活动信息 | 会议活动信息添加成功，跳转到会议活动信息管理页面并在会议活动信息列表中显示新增的会议活动信息 | 通过 |
| 3 | 在会议活动信息列表中点击修改，在会议活动信息信息页面修改完数据后单击提交 | 成功修改会议活动信息信息，提示信息为“操作成功”，跳转至会议活动信息管理页面 | 成功修改会议活动信息信息，提示信息为“操作成功”，跳转至会议活动信息管理页面 | 通过 |
| 4 | 在会议活动信息列表中点击删除 | 成功删除会议活动信息，并刷新所有会议活动信息页面 | 成功删除会议活动信息，并刷新所有会议活动信息页面 | 通过 |

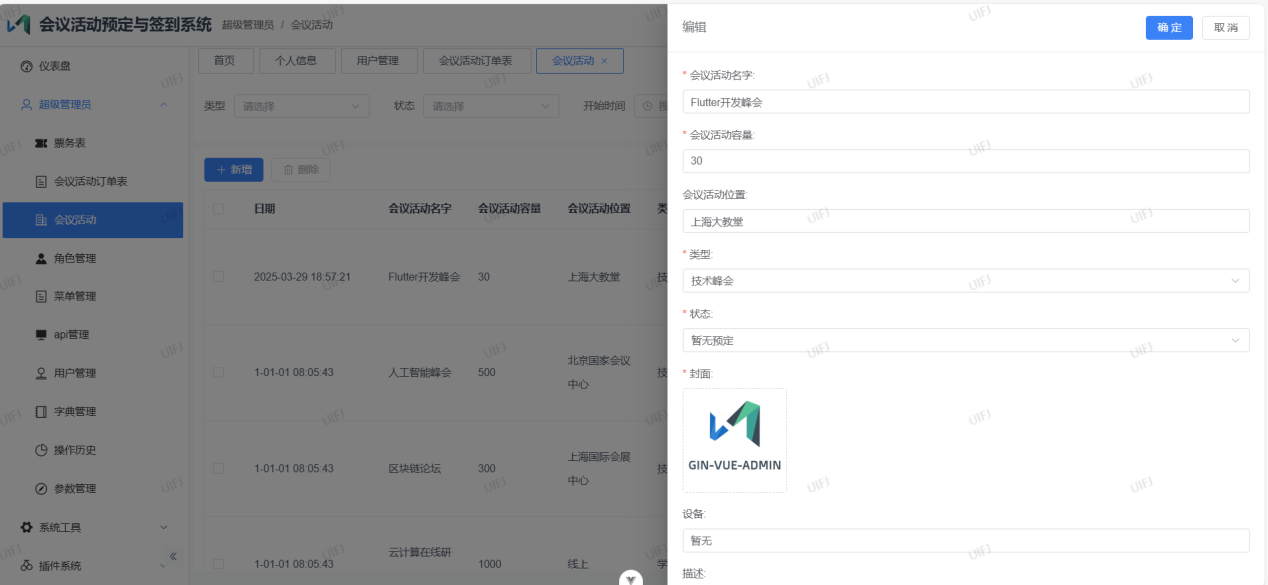


图5.4　会议活动信息管理功能测试结果图

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 |

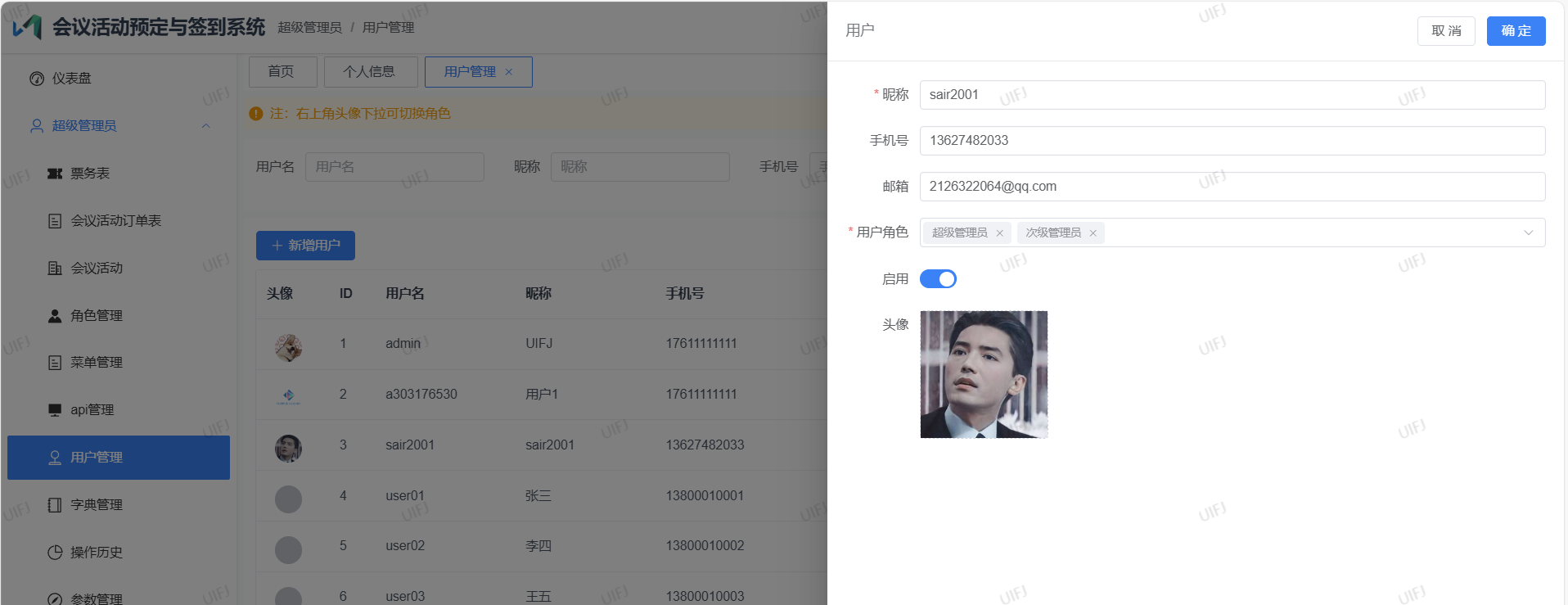


图5.5　用户管理功能测试结果图

## 5.3 测试结果分析

经过以上测试后对会议活动票务与签到管理系统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。

# 总 结

本研究基于Golang语言与Gin框架构建后端系统，结合Vue与Flutter框架开发前端应用，设计并实现了会议活动票务与签到管理系统。系统涵盖用户登录、会议活动管理、订单管理及票务管理等核心功能模块，支持用户线上完成会议预约与签到全流程，帮助运营者实现精准信息维护与高效用户管理，具备显著的应用价值。通过实际开发与测试验证，本系统在提升会议活动管理效率、优化资源利用率及改善用户体验方面取得了预期成果，为企业的数字化转型提供了可行的技术方案。

然而，受限于开发周期与技术能力，当前系统在功能深度与智能化水平上仍存在改进空间。例如，未来可通过引入AI客服实现实时问题解答与常见问题自动归纳，或集成实时消息推送功能，避免用户因信息滞后而错过关键会议活动。此外，系统的交互流程与多终端适配性有待进一步优化，以降低用户操作门槛。

展望未来，会议活动管理系统的升级应聚焦以下几个方向：

（1）智能化升级：结合物联网技术部署智能签到设备（如人脸识别终端或NFC签到），替代现有扫码方式，提升签到效率；扩展动态调价、智能推荐等功能，构建全链路服务生态。

（2）用户体验优化：精简界面逻辑，适配多终端（移动端、PC端及平板设备），探索VR/AR技术以增强线上会议的沉浸感。

（3）数据安全强化：采用区块链技术提升数据防篡改能力，完善分级权限管理与实时风险监测机制，确保用户隐私与交易安全。

通过上述改进，本系统有望成为会议管理领域的标杆解决方案，推动行业服务标准的提升，最终实现更高效、智能、安全的活动管理生态，为企业可持续发展注入新动能。

# 致 谢

回首论文撰写的艰辛历程与实习生涯的点滴积累，方觉时光如白驹过隙，转瞬即逝。站在毕业的门槛上，既有对过往岁月的感慨，亦怀揣对未来的期许。前路漫漫，唯愿永葆初心，脚踏实地，以赤子之心拥抱每一段征程。

首先，谨向我的指导老师\*\*教授致以最诚挚的谢意。无论是课堂上的倾囊相授，还是课后对学术研究与职业规划的悉心指导，钟老师始终以严谨的治学态度和无私的奉献精神为我点亮前行的明灯。每遇困顿，老师总能以渊博的学识与温暖的支持助我破局。常言道，“大学教师是学子一生所能遇见的最纯粹的知识引路人”，我何其有幸，在求学生涯中得遇如此良师。

其次，衷心感谢辅导员及各位专业老师的培育之恩。辅导员老师以耐心与智慧为我勾勒出多彩的大学生活图景，鼓励我参与各类实践活动，在团队协作与竞赛中锤炼能力、拓展视野；而各科任课教师则以深厚的专业积淀，授我以立足社会的技能，更以言传身教传递治学为人之道。得益于他们的指引，我不仅夯实了专业基础，更在学科竞赛中斩获佳绩，为青春留下了奋斗的注脚。

同时，感谢同窗挚友与舍友的陪伴与支持。在毕业设计攻坚阶段，他们以真诚的建议助我规避误区，以并肩作战的默契共克技术难题；在压力交织的深夜，我们彼此鼓舞、畅谈理想，让这段旅程少了几分孤独，多了几分温情。那些思维碰撞的火花与相互扶持的瞬间，终将沉淀为最珍贵的回忆。

最后，感恩家人的无私付出与默默守候。他们教会我在生活与学业间寻找平衡，以包容与理解为我筑起避风的港湾。无论顺境逆境，家人始终是我最坚实的后盾，赋予我直面挑战的勇气与笃定前行的力量。

岁月不居，未来已来。愿自己不畏年岁增长，不困于急功近利，以“白日梦想家”的热忱探索未知，以脚踏实地的姿态书写人生。愿此间所学所感，皆化为生命长河中的璀璨星光，照亮远方，亦温暖来路。

# 参考文献

1. 王畅.基于BS架构的应用系统国产化环境适配策略及实践[J].网络安全技术与应用,2024, (04):17-20.
2. I. O. Joshua, M. O. Arowolo, M. O. Adebiyi, K. A. Gbolagade, J. O. Olaniyan and E. T. Aderemi, "Design and Implementation oF a Single Sign-In Identity (SSID) oN tHE Banking System Using a Blockchain Algorithm," 2024 International Conference on Science, Engineering and Business for Driving Sustainable Development Goals (SEB4SDG), Omu-Aran, Nigeria, 2024, pp. 1-12, doi: 10.1109/SEB4SDG60871.2024.10630174.
3. A. A. Santoso, M. Y. S. A. Winata, M. L. A. Cahyadi and I. Revelino, "Approval System Development in Information System for Project Management in a Retail Company," 2024 International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA), Bali, Indonesia, 2024, pp. 401-406, doi: 10.1109/ICICYTA64807.2024.10912934.
4. Z. Li, "Automatic Student Sign in Management System Based on Neural Network Algorithm," 2022 International Conference on Artificial Intelligence of Things and Crowdsensing (AIoTCs), Nicosia, Cyprus, 2022, pp. 121-125, doi: 10.1109/AIoTCs58181.2022.00025.
5. Z. Jiang, Z. Si and C. Luo, "Design & Implementation to an RFID Based Conference Management System," 2014 Tenth International Conference on Computational Intelligence and Security, Kunming, China, 2014, pp. 143-147, doi: 10.1109/CIS.2014.29.
6. 王志强, 刘洋. 移动互联网时代会议活动管理系统的创新与发展[J]. 现代计算机, 2023, 39(05): 45-49. DOI:10.3969/j.issn.1007-1423.2023.05.008.
7. 杨帆, 孙晓东. 数据驱动的会议活动管理系统设计与实现[J]. 软件导刊, 2023, 22(03): 123-127. DOI:10.3969/j.issn.1672-7808.2023.03.025.
8. 陈宇涵. 会议室综合管理系统研究与开发[D]. 四川:电子科技大学,2023.
9. 刘慧玲,谭定英,陈平平.基于SpringBoot和Vue.js的大学生团队管理系统的设计[J].电脑编程技巧与维护,2024,(03):120-122.DOI:10.16184/j.cnki.comprg.2024.03.039.
10. 沈慧.智能视频会议系统的设计与实现[J].集成电路应用.2023,40(9).DOI:10.19339/j.issn.1674-2583.2023.09.117 .
11. 冯世诚,李彬,林文彬.无纸化会议系统的实践与思考[J].铁道运营技术.2023,29(1).48-50.DOI:10.13572/j.cnki.tdyy.2023.01.014 .
12. 赵静, 李伟. 智能化票务管理系统的功能优化与用户体验提升研究[J]. 计算机科学, 2022, 49(S1): 312-316. DOI:10.11896/jsjkx.220100115.
13. 张明辉, 李晓峰. 基于云计算的会议活动票务管理系统设计与实现[J]. 计算机工程与应用, 2022, 58(12): 78-83. DOI:10.3969/j.issn.1002-8331.2022.12.012.
14. 时明.Web主流前端开发框架研究[J].信息记录材料,2020,21(05):215-216.DOI:10.16009/j.cnki.cn13-1295/tq.2020.05.143.
15. 秦溧,艾青.基于Redis技术的页面缓存研究[J].科技风.2021,(12).DOI:10.19392/j.cnki.1671-7341.202112035 .
16. 罗超.基于java的会议活动票务与签到管理系统关键技术研究[J].信息记录材料,2019,20(08):159-160.DOI:10.16009/j.cnki.cn13-1295/tq.2019.08.105.
17. 潮洛蒙.基于B/S的大型企业智能会议管理系统的设计与实现[D].吉林省:吉林大学,2016.