基于 springboot+vue 的旅游管理系统的设计与实现

作者：***

目　　录

第一章 系统概述

随着蓬勃发展的旅游业和日益普及的互联网技术，越来越多的旅行者开始依赖便捷高效的在线平台来规划他们的精彩纷呈的旅程。在线旅游信息推荐系统，作为一种智能化的信息服务工具，正是为了帮助用户轻松而精准地找到最契合心意的旅游景点而精心设计的。通过高度个性化的智能推荐，系统能够细腻地匹配用户的独特兴趣和实际需求，大幅减少他们在浩瀚如海的旅游信息中反复筛选的时间和精力，从而显著提升整体旅行体验和深层次的满意度。

一套真正高效的推荐系统，不仅能引导游客探索新奇独特的小众景点，还能促进旅游资源的科学合理分配和均衡可持续的发展，同时有力刺激旅游消费，推动整个行业的繁荣与创新。此外，通过持续提供高度契合用户期望的推荐内容，系统能够不断增强用户对平台的信赖感与依赖度，提高用户的长期留存和活跃互动，为平台创造源源不断的流量和持久的商业价值。

不仅如此，基于海量用户行为数据的深度分析，系统还能敏锐洞察旅游市场的最新趋势和热门动向，为旅游企业、地方政府及相关机构提供强有力的决策依据，助力优化旅游资源的高效配置与合理开发。旅游景点推荐系统的实现，涉及前沿的机器学习、精准的数据挖掘、智能化的自然语言处理等多领域技术，其研发与应用不仅推动了行业技术的革新，也为未来智慧旅游的发展奠定了坚实基础。

该系统旨在通过细致入微地分析用户的个性化偏好、历史行为等数据，为用户提供丰富多彩的旅游景点推荐，满足不同群体的多元化需求。同时，通过推荐高品质、独具特色的旅游目的地，系统能够引导用户进行更高层次的消费升级，提升整体旅游体验的精致度与满意度。

综上所述，旅游景点推荐系统的实现，对于全面提升用户体验、加速旅游业高质量发展、增强用户忠诚度、优化行业资源配置以及促进技术创新突破等方面均具有不可忽视的重要意义。其最终目标在于精准满足用户需求、提高推荐智能化水平、增强用户幸福感与归属感、推动旅游消费品质升级，并最终实现商业价值与社会效益的双赢。

第二章 系统需求分析

2.1 任务概述

这套旅游推荐系统的功能模块设计精心考量了用户和管理员的双重需求，巧妙实现了丰富多样、细致入微的功能体系。在用户模块方面，系统不仅提供了便捷高效的景点浏览、灵活多样的购票方式，还囊括了个性化定制的旅游线路推荐、详尽全面的酒店信息查询以及实时更新的旅游资讯推送。此外，人性化的注册与登录流程、贴心实用的“我的收藏”功能，进一步为用户打造了专属化、智能化的服务体验，让每一次旅行规划都轻松愉悦、省心省力。

管理员模块则采用了科学严谨、条理分明的设计思路，涵盖了高效精准的账号管理、灵活可控的线路管理、动态优化的景点信息管理、实时追踪的订单信息管理以及内容精细的资讯管理功能。这些模块不仅大幅提升了管理员的工作效率，还有力支撑了旅游业务的规范化、数字化运营，推动行业向智能化、现代化方向稳步迈进。

整体而言，该系统通过逻辑清晰、层次分明的功能模块划分，完美兼顾了用户与管理员的使用需求，不仅让用户能够轻松自如地达成旅行目标，也让管理员得以游刃有余地开展管理工作。在流畅稳定的系统运行和优质贴心的用户体验双重保障下，这套系统必将成为旅游行业数字化转型的标杆之作，为行业的可持续发展注入强劲动力。

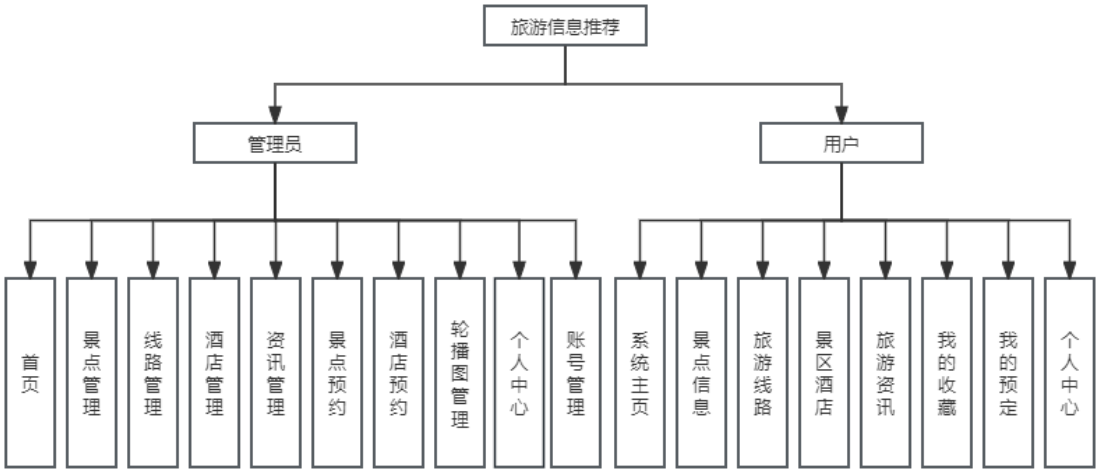

图 2.1 系统功能概述

2.2 功能性需求

旅游信息推荐系统主要分为两大功能模块：系统管理员模块和用户使用端模块。

管理端模块主要负责整个系统的配置及系统安全运行保证。管理端模块包括首页、景点管理、线路管理、酒店管理、资讯管理、景点预约、酒店预约、用户管理、轮播图管理、个人中心

首页，包括最近七日酒店预约和景点预约的可视化图表分析。

景点管理模块，包括对景点的增加、删除、修改、查询功能。

线路管理模块，包括对线路的增加、删除、修改、查询功能。

酒店管理模块，包括对酒店的增加、删除、修改、查询功能。

资讯管理模块，包括对资讯的增加、删除、修改、查询功能。

景点预约，查看用户的预约订单。

酒店预约，查看用户的预约订单。

用户管理，对管理员和用户的账号管理，可以修改账号信、重置密码和删除。

个人中心，对管理员自身信息的修改，包括密码和头像的修改。

学生模块主要包含了系统主页、景点信息、旅游线路、景区酒店、旅游资讯、我的收藏、我的预定、个人中心等功能。

2.3 非功能性需求

为了保证系统能够长期、安全、稳定、可靠、高效的运行，系统应该满足以下的性能需求：

2.3.1 正确性需求

管理员应能够进行有关的用户信息准确地添加到数据库中。系统用户登录后，系统应能正确地读取用户个人信息以及用户的权限信息系统的操作结果与预期的结果应该是一致。

2.3.2 安全性需求

系统登录需要输入用户名、密码，并且需要防止 SQL 注入问题，用户登录后根据用户角色的不同可以访问的权限应该不同。

2.3.3 界面需求

系统对界面的要求窗口布局清晰，颜色搭配合理，色调柔和，各窗体主题风格一致，对用户友好，界面的设计应遵循如下规则：

（1）提供信息反馈，用多种信息提示用户当前的系统运行状态。

（2）显示启动画面，画面要简洁明了，不能太过花哨。

2.3.4 时间特殊性需求

当管理员向系统添加用户信息时需要在一定时间内处理用户数据并将数据录入数据库中。

2.3.5 稳定性需求

系统部署后，在硬件条件和支持软件条件没有变化的情况下，能够一直保持运行状态，直到系统被升级或代替。

2.3.6 故障处理能力需求

系统可能遇到的软件故障是数据库与应用程序服务器。为了满足信息处理的需求，可以采取数据恢复数据来解决。

2.4 开发技术简介

2.4.1 开发工具简介

IDEA 全称 IntelliJ IDEA，是 java 编程语言开发的集成环境。IntelliJ 在业界被公认为最好的 java 开发工具，尤其在智能代码助手、代码自动提示、重构、JavaEE 支持、各类版本工具(git、svn 等)、JUnit、CVS 整合、代码分析、 创新的 GUI 设计等方面的功能可以说是超常的。IDEA 是 JetBrains 公司的产品，这家公司总部位于捷克共和国的首都布拉格，开发人员以严谨著称的东欧程序员为主。它的旗舰版本还支持 HTML，CSS，PHP，MySQL，Python 等。免费版只支持 Java，Kotlin 等少数语言。2001 年 1 月发布 IntelliJ IDEA 1.0 版本，同年七月发布 2.0，接下来基本每年发布一个版本(2003 除外)，当然每年对各个版本都是一些升级。3.0 版本之后，IDEA 屡获大奖，其中又以 2003 年的赢得的“Jolt Productivity Award”，“JavaWorld Editors’ s Choice Award”为标志，从而奠定了 IDEA 在 IDE 中的地位。

2.4.2 开发技术简介

1.SpringBoot

Spring Boot 是由 Pivotal 团队精心打造的革命性框架，其核心理念在于大幅简化新一代 Spring 应用的初始化搭建与开发流程。该框架采用了高度智能化的配置方式，通过约定优于配置（Convention Over Configuration）的先进理念，使开发人员彻底摆脱了传统开发中繁琐重复的样板化配置工作。

Spring Boot 通过自动化配置（Auto-Configuration）、内嵌服务器（Embedded Server）以及起步依赖（Starter Dependencies）等创新特性，让开发者能够快速聚焦于业务逻辑的实现，而非底层架构的搭建。这种高效便捷的开发模式，不仅显著降低了项目的启动成本，还极大提升了整体开发效率，使得 Spring Boot 在日新月异的快速应用开发（Rapid Application Development, RAD）领域迅速崛起，成为当之无愧的行业标杆。

此外，Spring Boot 还深度整合了 Spring 生态系统的丰富功能，同时保持了极佳的灵活性，无论是构建轻量级微服务，还是开发复杂的企业级应用，都能游刃有余地应对。其强大的可扩展性和卓越的性能表现，使其成为现代 Java 开发者的首选框架，持续引领着企业级应用开发的技术潮流。

2.MYSQL 数据库介绍

MySQL 是一个高效稳定的关系型数据库管理系统（RDBMS），最初由瑞典 MySQL AB 公司精心研发，目前归属于全球知名的 Oracle 公司旗下。MySQL 采用先进的关联式数据库架构，通过将数据科学拆分并存储在不同的表中，而非简单地堆积在一个庞大的数据仓库内，从而显著优化了数据查询和管理的运行效率与系统灵活性。

作为一款开源免费的数据库软件，MySQL 凭借其卓越的性能、可靠的稳定性和出色的扩展性，已成为全球最受欢迎的数据库解决方案之一。它广泛应用于各类 Web 应用、企业级系统和云计算平台，为开发者提供了强大而灵活的数据存储与管理能力。无论是小型网站还是高并发的互联网服务，MySQL 都能游刃有余地应对，持续为现代数据管理提供坚实的技术支撑。

本系统使用 MySQL 作为开发中使用的数据库，它具有使用简单，稳定等特性。在与 java 程序连接时，为提到数据库操作的效率提高系统的性能。使用到 DBUtils 和 DBCP 等工具。

在使用 DBUtils 之前，我们 Dao 层使用的技术是 JDBC，那么分析一下 JDBC

的弊端。数据库链接对象、sql 语句操作对象，封装结果集对象，这三大对象会重复定义封装数据的代码重复，而且操作复杂，代码量大。释放资源的代码重复。

3.Vue

Vue 是由华人开发者尤雨溪（Evan You）于 2013 年（与 React 框架同年问世）推出的一个革命性的渐进式、自底向上的前端框架。作为现代前端开发的标杆之作，Vue 以其优雅的设计理念和出色的开发体验迅速赢得了全球开发者的青睐。

那么，什么叫做渐进式框架呢？从专业角度来说，渐进式框架意味着开发者可以以 Vue 的核心库为基础，随着项目需求的不断深入和业务规模的扩大，逐步引入其丰富的生态系统（如 vue-router 实现路由管理、vuex 进行状态管理、nuxt.js 实现服务端渲染等），从而实现对项目的深度定制和渐进增强。

用更通俗易懂的话来说，Vue 允许开发者像"搭积木"一样，从最简单的功能开始，逐步扩展项目规模。我们可以先使用 Vue 的基础功能来优化项目中的表单提交和列表渲染，随着业务发展再引入路由系统构建多页面应用，最终甚至可以扩展到服务端渲染（SSR）等高级特性。这种循序渐进的采用方式，让开发者能够根据实际需求灵活选择技术方案，既避免了过度设计，又能确保项目的可扩展性和长期可维护性。

Vue 的渐进式哲学不仅降低了前端开发的入门门槛，也为项目演进提供了平滑的升级路径，这正是它能在众多前端框架中脱颖而出的关键所在。其轻量级的核心库（仅 23kb gzip 后）、直观的模板语法和响应式的数据绑定机制，共同构成了一个高效而优雅的前端开发解决方案。

结果：（1）程序员在开发的时候，有大量的重复劳动。（2）开发的周期长，效率低。数据库连接是一种关键的有限的昂贵的资源，这一点在多用户的网页应用程序中体现的尤为突出.对数据库连接的管理能显著影响到整个应用程序的伸缩性和健壮性，影响到程序的性能指标.数据库连接池正式针对这个问题提出来的.数据库连接池负责分配，管理和释放数据库连接，它允许应用程序重复使用一个现有的数据库连接，而不是重新建立一个。数据库连接池在初始化时将创建一定数量的数据库连接放到连接池中， 这些数据库连接的数量是由最小数据库连接数来设定的.无论这些数据库连接是否被使用，连接池都将一直保证至少拥有这么多的连接数量.连接池的最大数据库连接数量限定了这个连接池能占有的最大连接数，当应用程序向连

接池请求的连接数超过最大连接数量时，这些请求将被加入到等待队列中。

在软件开发过程中，数据库连接管理是一个至关重要的性能瓶颈问题。首先从开发效率角度来看，传统数据库连接方式存在明显缺陷。每当应用程序需要访问数据库时，都需要经历建立连接、身份验证、权限检查等一系列复杂过程。这种重复性的连接操作不仅造成了程序员大量的重复劳动，还显著延长了开发周期，降低了整体开发效率。特别是在多用户并发的 Web 应用场景中，这种低效的连接方式会导致系统资源被严重浪费。

其次从系统性能角度分析，数据库连接是一种极其宝贵的有限资源。每个连接都需要占用服务器内存、CPU 等计算资源，同时还会产生网络开销。在高并发环境下，频繁地创建和销毁连接会导致系统性能急剧下降，甚至可能引发连接耗尽的问题。这种资源竞争状况会直接影响应用程序的响应速度和稳定性。

数据库连接池技术正是针对这些痛点提出的创新解决方案。其核心工作原理包括资源预分配机制、连接复用机制、动态扩容能力和智能管理功能。连接池在初始化阶段就会创建一定数量的数据库连接，这些连接会一直保持活跃状态，确保随时可用。应用程序可以从连接池中获取现有的空闲连接，使用完毕后归还给连接池，这种复用模式大大减少了系统开销。当并发请求增加时，连接池会根据配置动态扩展连接数量，请求超过最大连接数时采用队列机制缓冲处理。

从技术实现层面来看，优秀的连接池解决方案通常会采用多种优化策略，包括使用轻量级的并发数据结构、实现高效的连接获取/释放算法、采用智能的连接有效性校验机制等。在实际应用中，合理配置连接池参数至关重要，需要根据服务器硬件配置和应用特点进行调优。最小连接数、最大连接数、连接超时时间等参数都需要根据具体业务场景进行优化配置。

从系统架构角度看，连接池技术的引入带来了显著优势，包括降低数据库访问延迟、提高系统吞吐量、增强应用程序伸缩性、改善系统稳定性等。特别是在云原生和微服务架构盛行的今天，数据库连接池已经成为现代应用开发不可或缺的基础组件。它不仅解决了传统连接方式的性能瓶颈，还为系统的高可用性提供了有力保障。通过合理使用连接池技术，开发团队可以在保证系统性能的同时，大幅提升开发效率，实现高效开发与高性能运行的完美结合。

第三章 系统总体设计

3.1 系统总体功能设计

考虑到本系统主要是的使用者是游客，考虑到游客的硬件设备和软件设备不足、学生的学习能力水平等因素，本系统的主要流程图如下图所示。

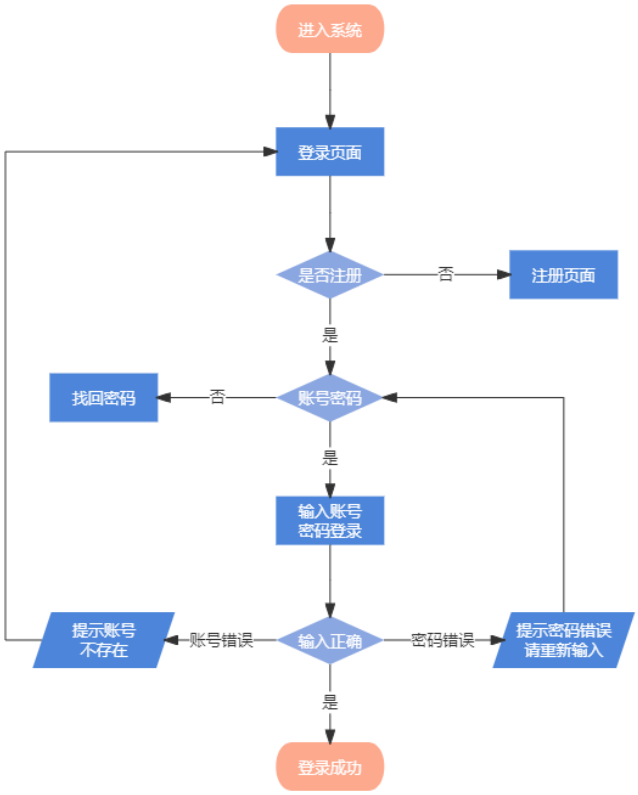

图 3.1.1 系统登录功能流程图

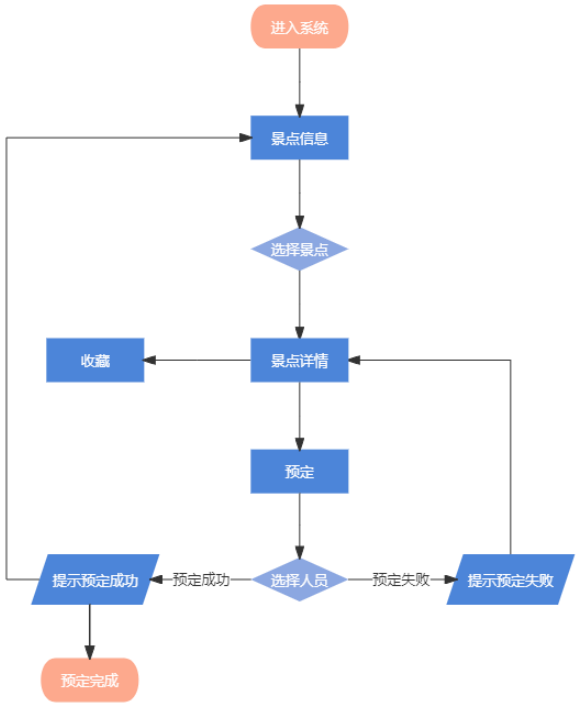

图 3.1.2 系统景区预定流程图

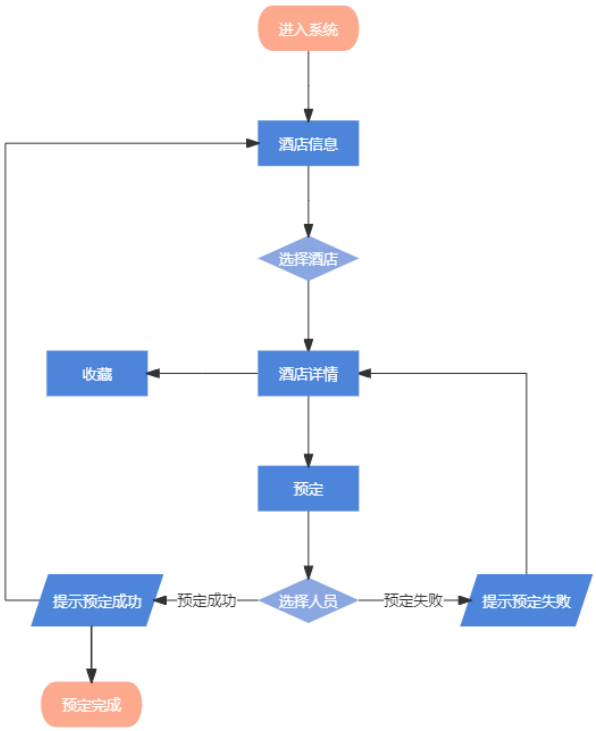

图 3.1.3 系统酒店预定流程图

3.2 系统数据库设计

3.2.1 概念结构设计

数据库的角色和表关系图如下图所示：

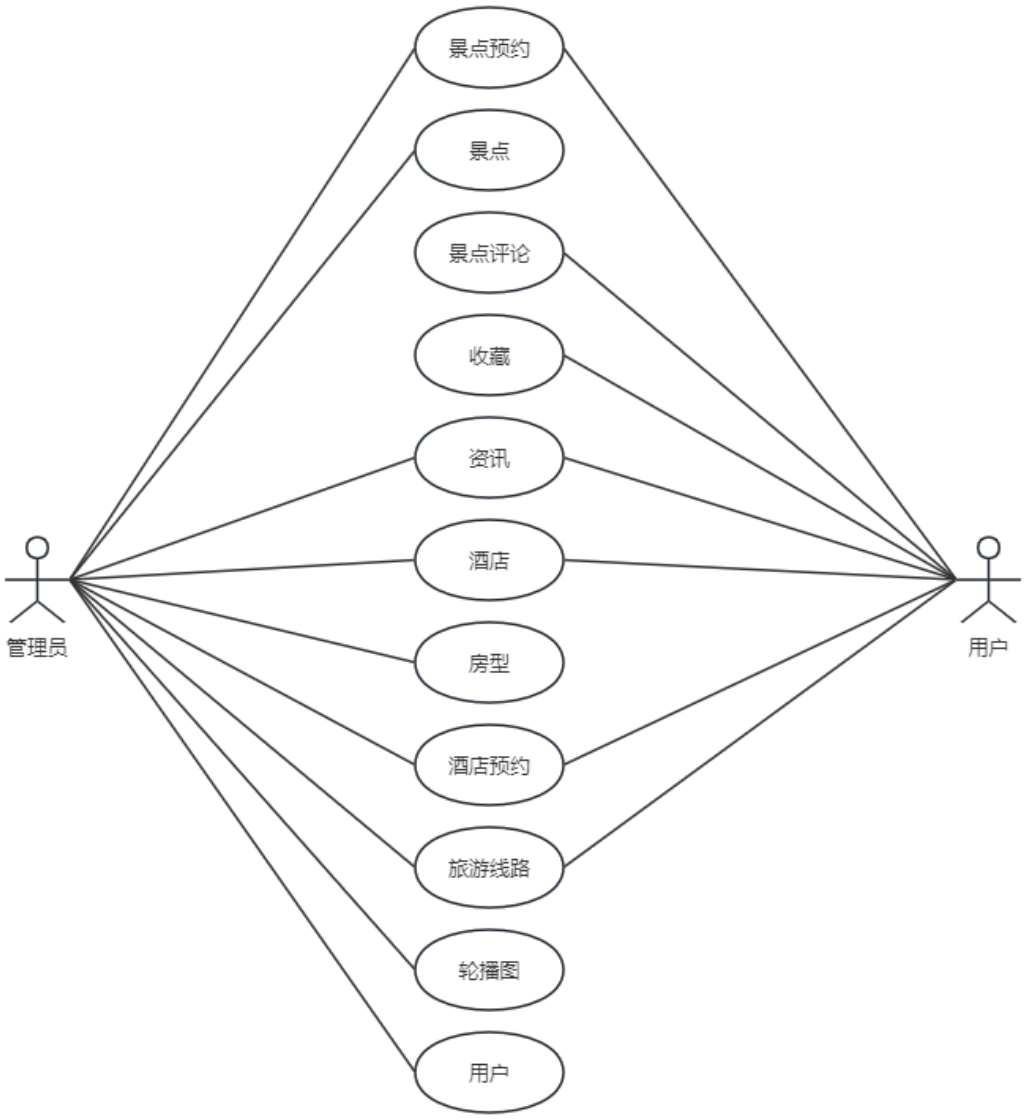

图 3.2 角色和表关系图

3.2.2 数据库逻辑结构设计

数据库的逻辑结构设计主要分为两步：第一步将概念设计模型得出的 E-R 图进行转换成关系模型，然后对转化成的关系模型进行优化。根据角色和表关系图可以确定系统设计 11 个数据表比较合适，分别是景点预约表（sys_attraction_order）、景点表（sys_attractions）、景点评论表（sys_comments）、收藏表（sys_favor）、资讯表（sys_forum）、酒店表（sys_hotel）、房型表（sys_hotel_item）、酒店预约表

（sys_hotel_order）、旅游线路表（sys_line）、轮播图表（sys_rotations）、用户表（user）。在系统的实体类图确定之后，需要将概念层进一步具体化。实体类之间的联系可以根据具体的情况确定。

具体的数据库 ER 图如下图所示：

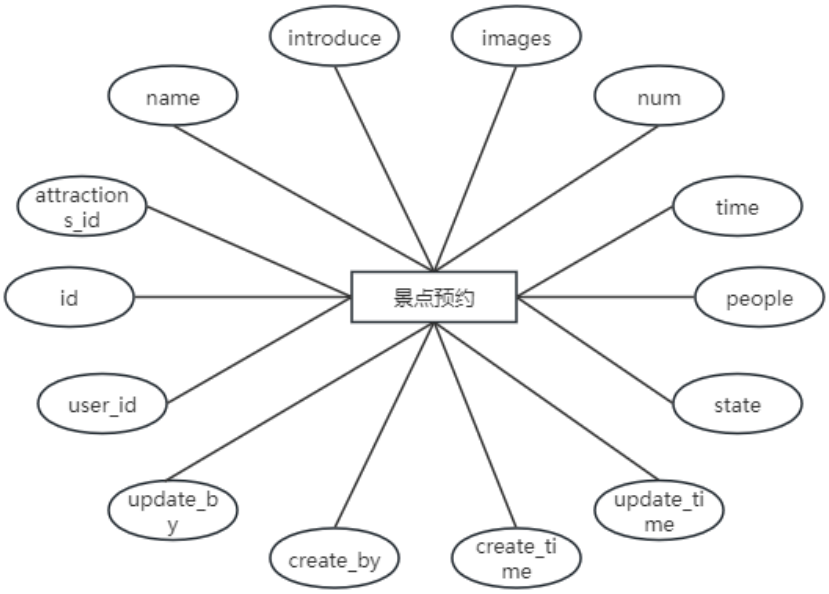

图 3.2.1 景点预约 ER 图

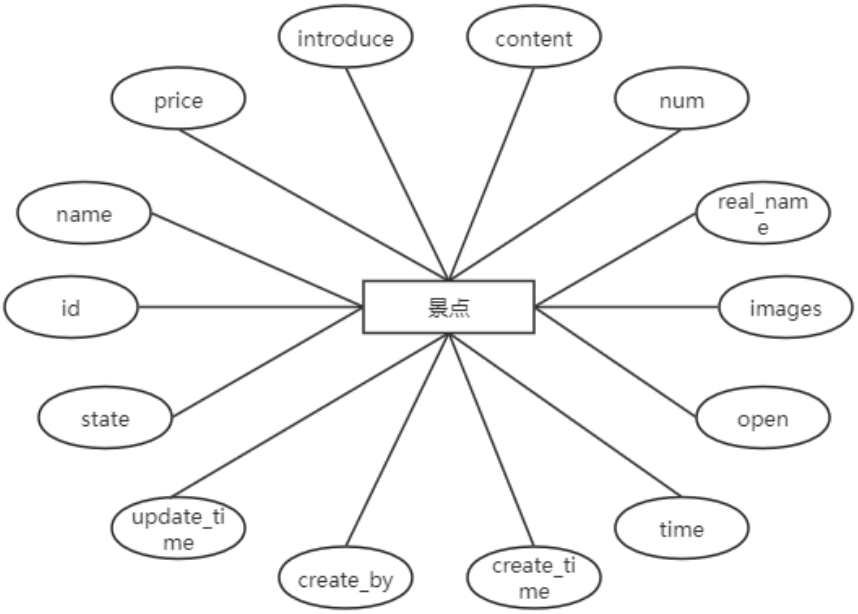

图 3.2.2 景点 ER 图

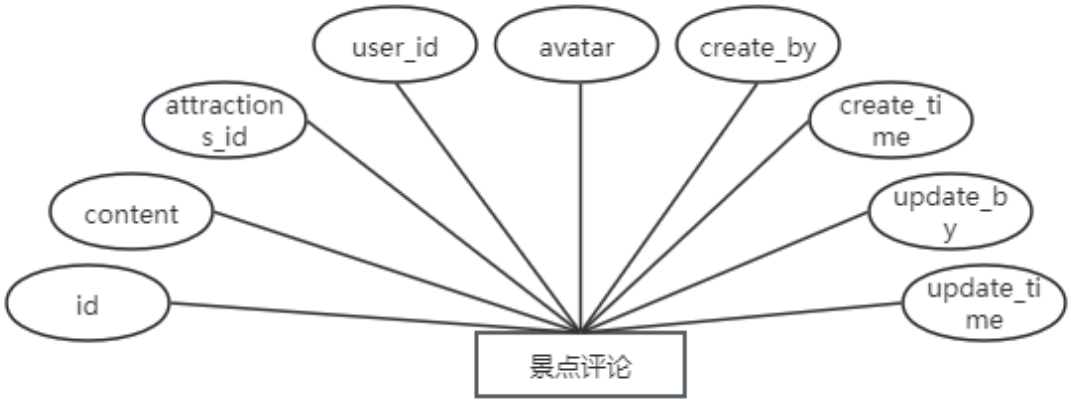

图 3.2.3 景点评论 ER 图

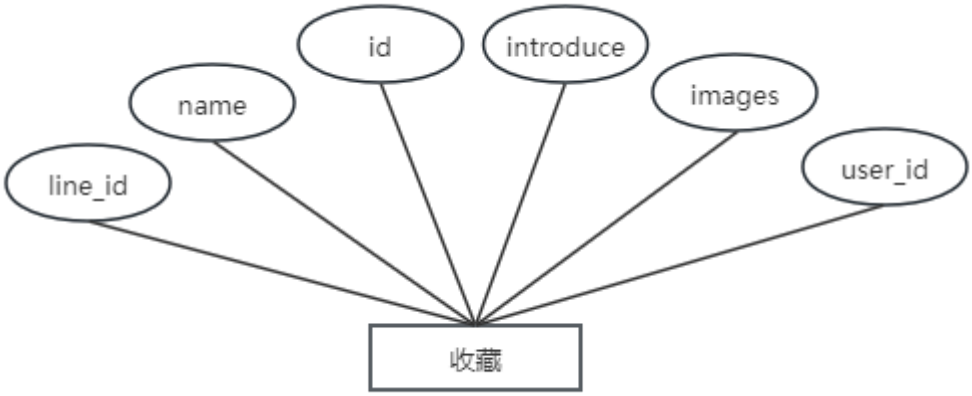

图 3.2.4 收藏 ER 图

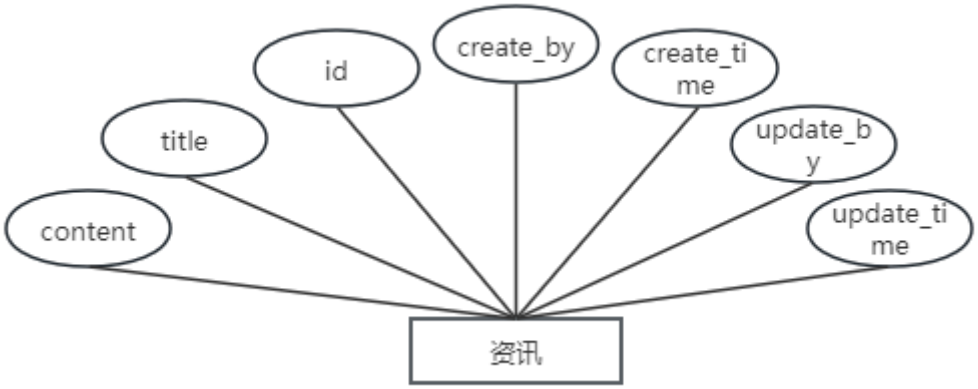

图 3.2.5 资讯 ER 图

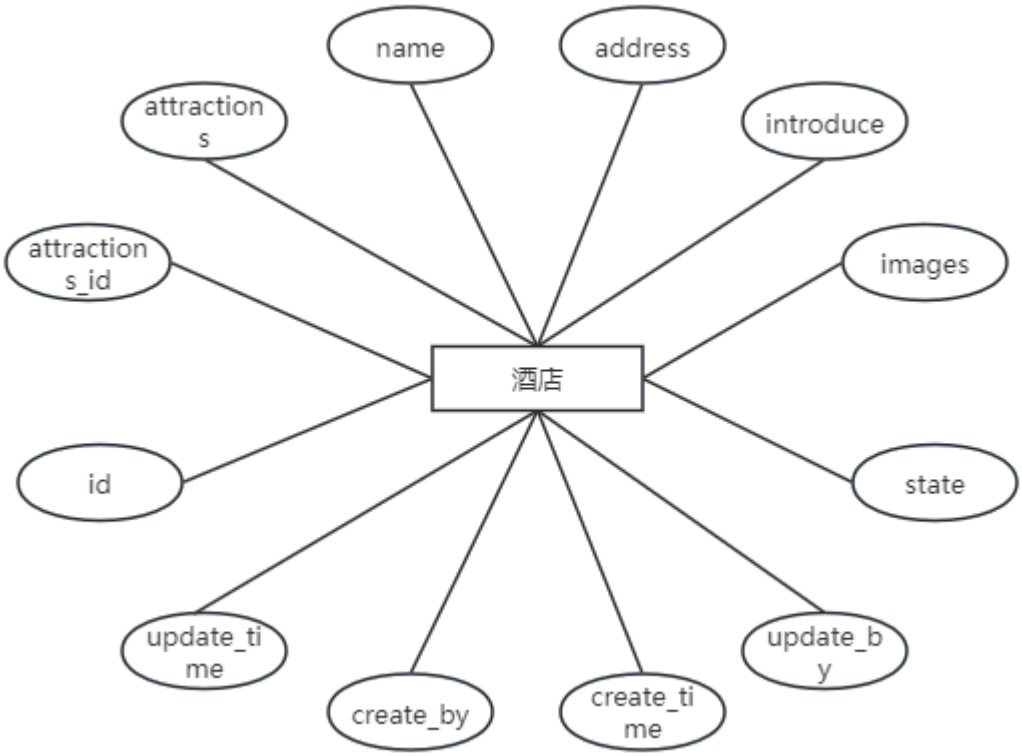

图 3.2.6 酒店 ER 图

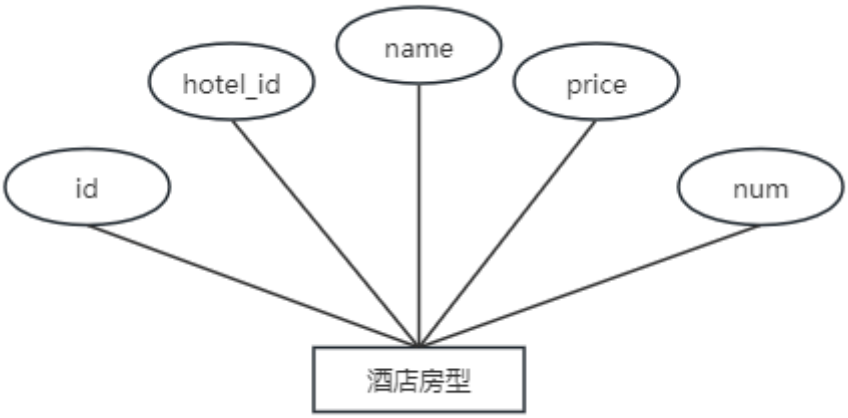

图 3.2.7 酒店房型 ER 图

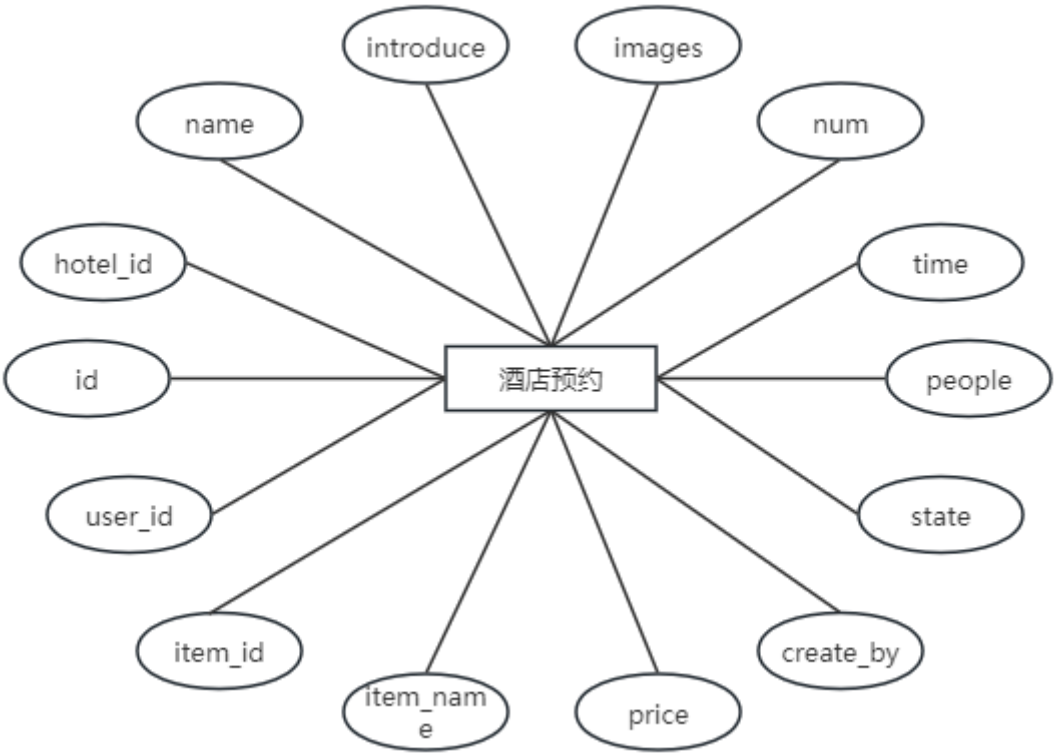

图 3.2.8 酒店预约 ER 图

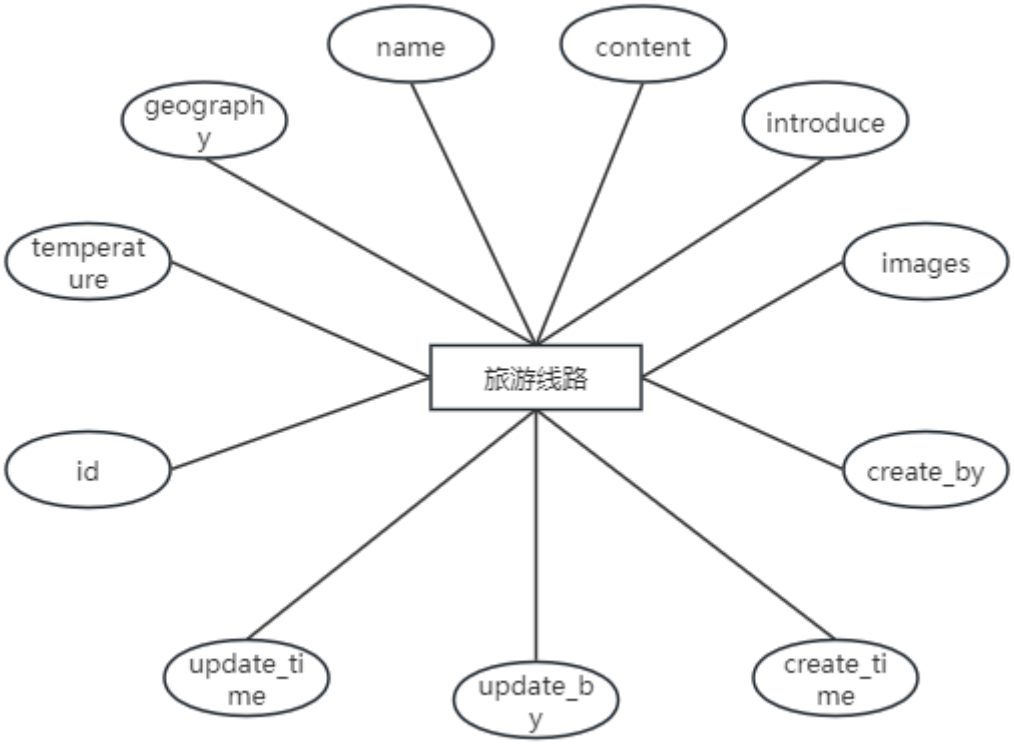

图 3.2.9 旅游线路 ER 图

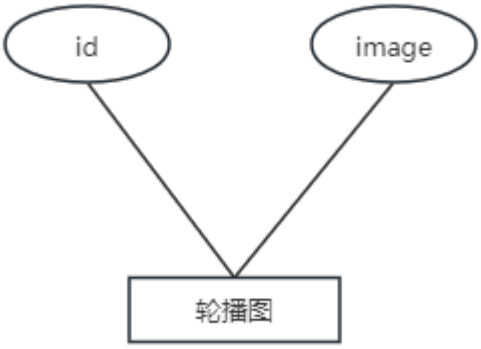

图 3.2.10 轮播图 ER 图

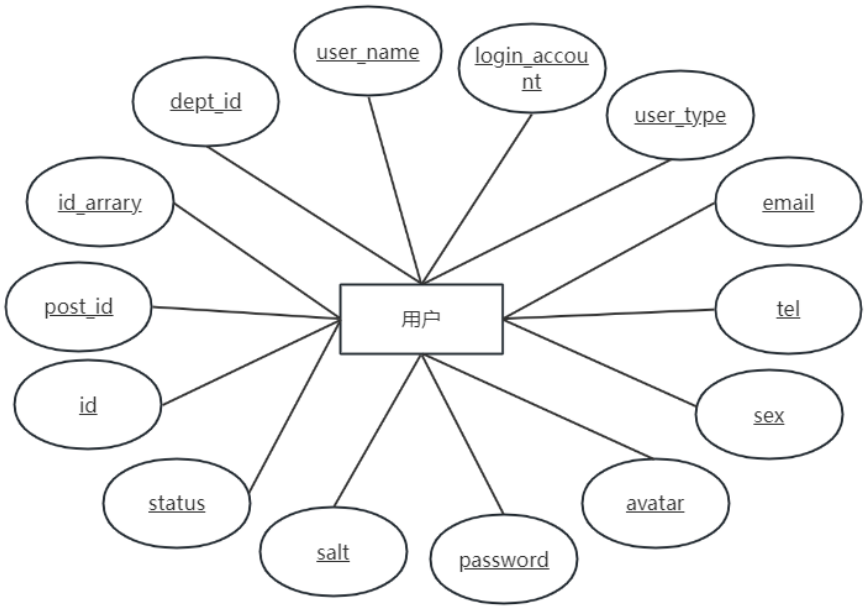

图 3.2.11 用户 ER 图

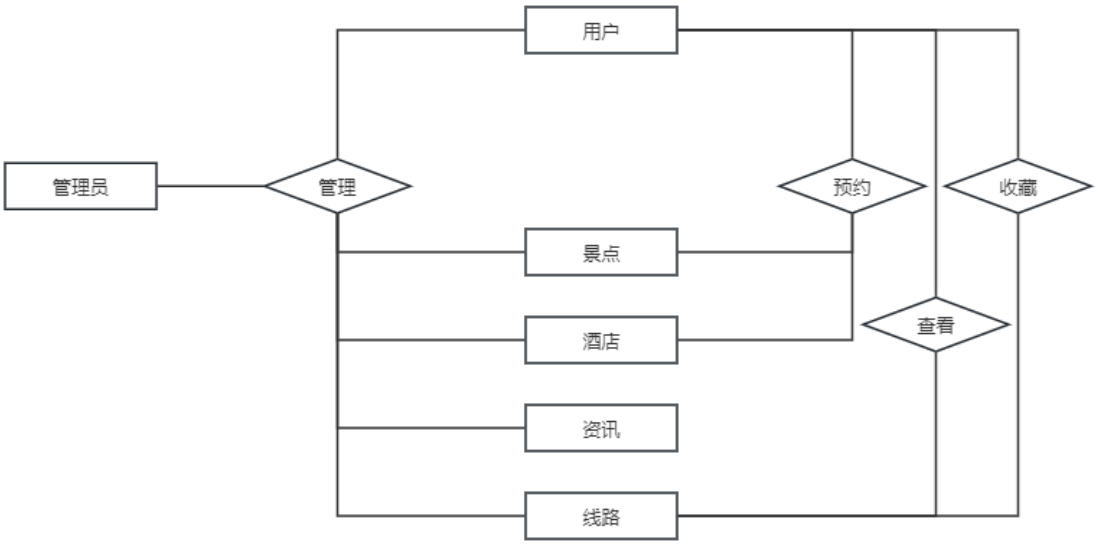

图 3.2.12 系统总体 ER 图

第四章 系统主要功能设计及实现

4.1 用户功能

用户在登录页面通过输入用户名和密码后点击登录按钮进行登录，用户的登录功能设计如下图所示：

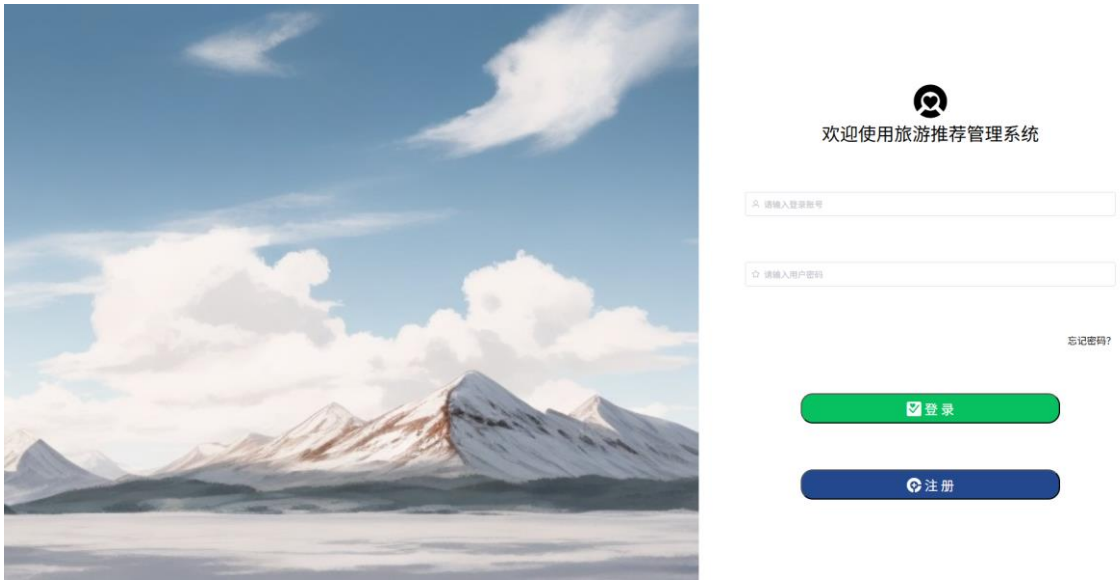

图 4.1.1 用户登录界面

用户在注册页面通过输入用户名、用户账号、密码和确认密码后点击注册按钮进行注册，用户的注册功能设计如下图所示：

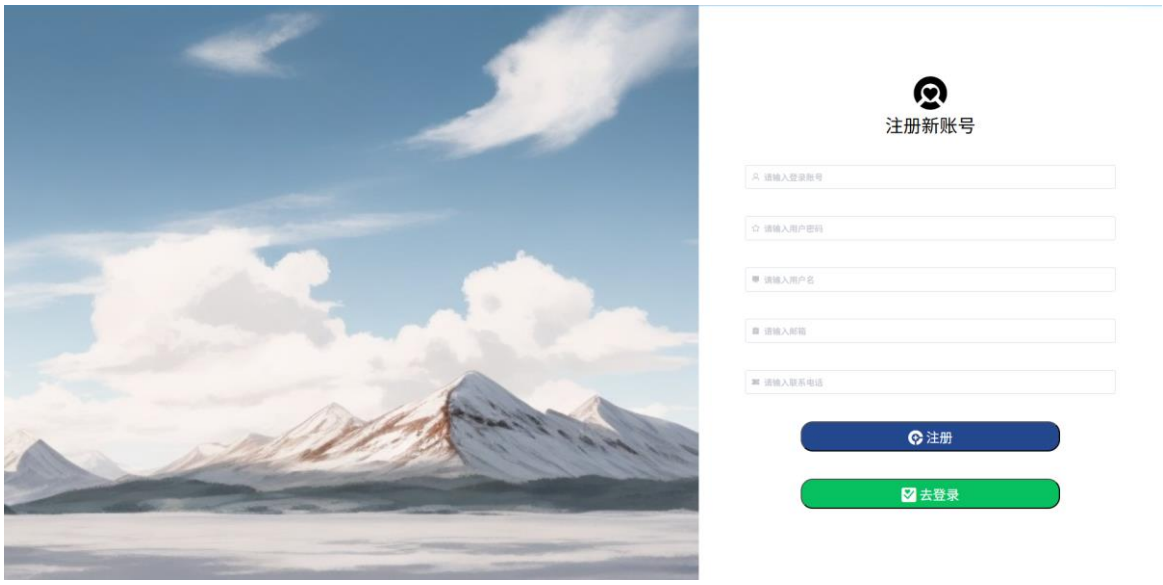

图 4.1.2 用户注册界面

用户在登陆成功后会进入到系统首页，系统首页展示了轮播图、推荐景点、平台统计数据、推荐线路，用户的首页功能设计如下图所示：

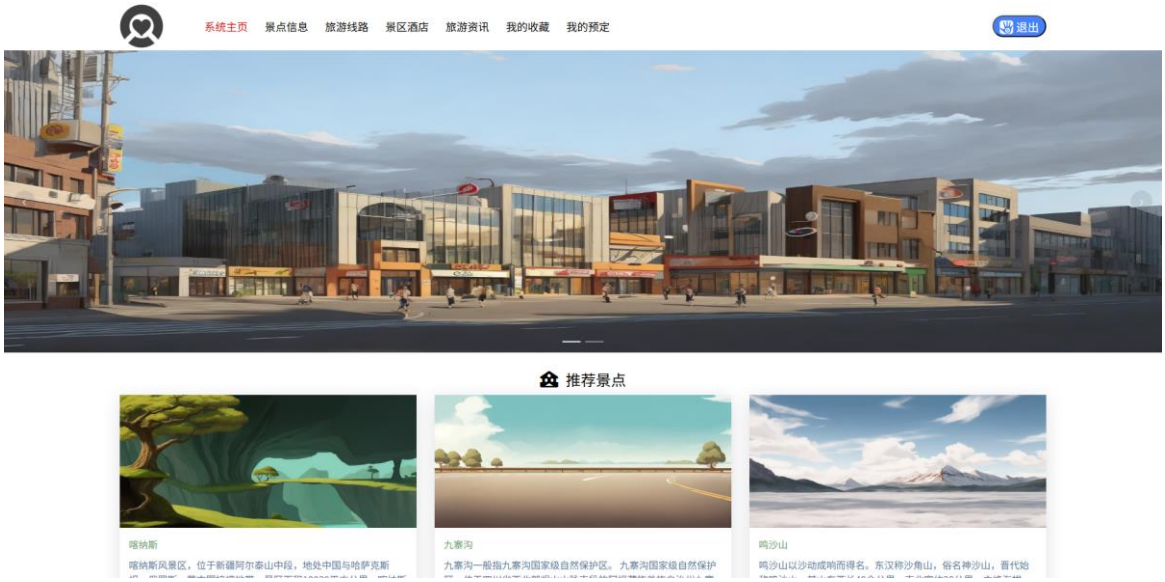
系统主页
景点信息
旅游线路
景区酒店
旅游资讯
我的收藏
我的预定
退出
推荐景点
喀纳斯
喀纳斯风景区，位于新疆阿尔泰山中段，地处中国与哈萨克斯
九寨沟
九寨沟一般指九寨沟国家级自然保护区。九寨沟国家级自然保护
鸣沙山
鸣沙山以沙动成响而得名。东汉称沙角山，俗名神沙山，晋代始

图 4.1.3 用户首页界面

点击导航栏的景点信息进入景点列表，可以根据景点名称检索，用户的景点列表功能设计如下图所示：

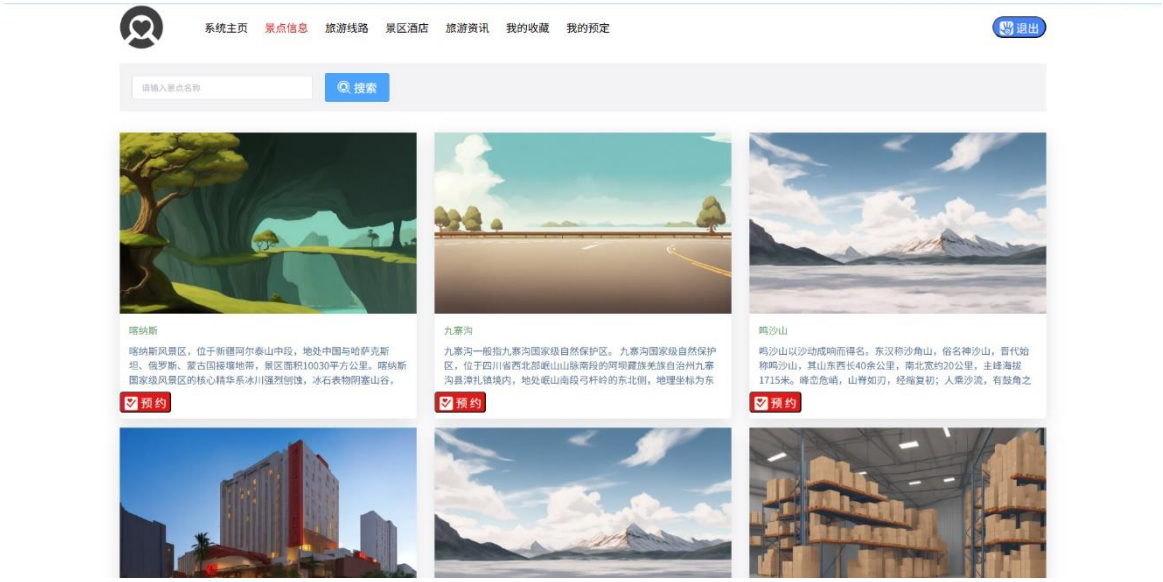

图 4.1.4 用户景点列表界面

进入景点列表之后，点击想要预约的景点，进入到景点详情页，景点详情页包含了景点名称、价格、简介、库存、介绍和评论功能，用户的景点详情功能设计如下图所示：

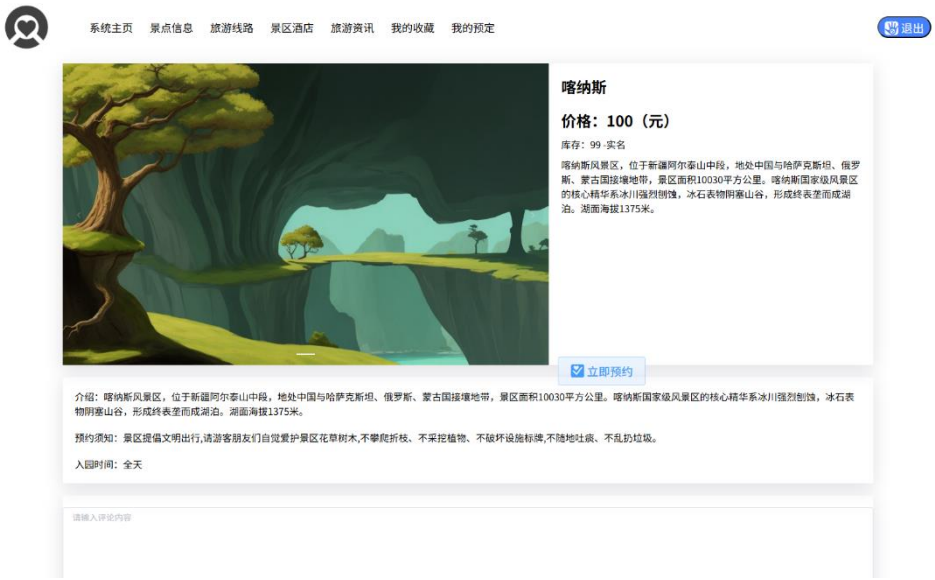

图 4.1.5 用户景点详情界面

点击立即预约，选择预约日期和填写预约人，如下图所示：

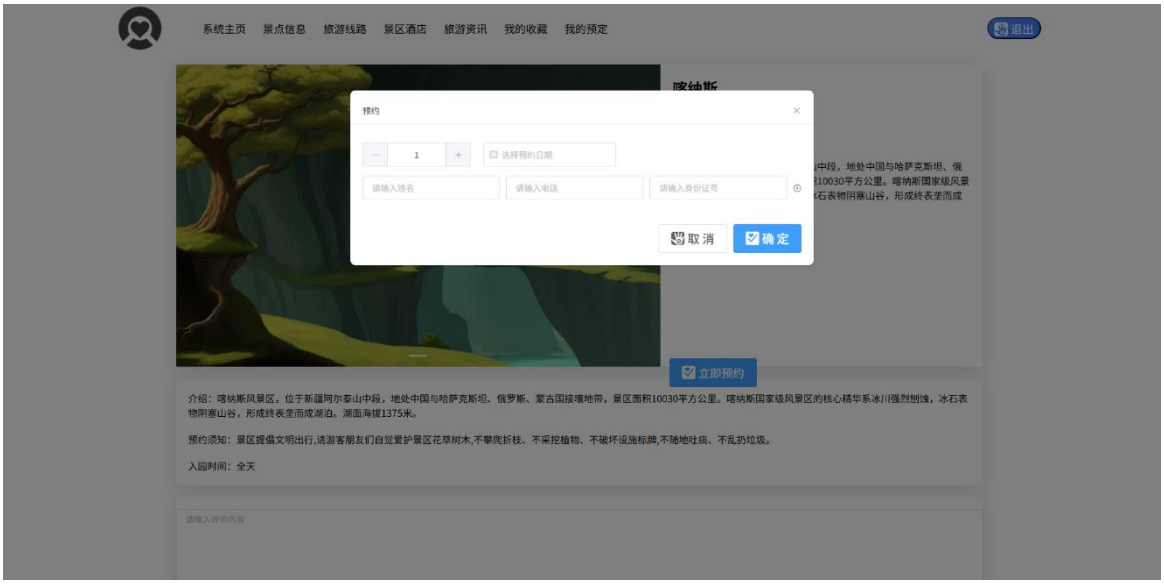

图 4.1.6 用户预约景点界面

点击导航栏的旅游线路进入到旅游线路列表，可以根线路名称、地理情况和温度进行筛选，旅游线路列表功能设计如下图所示：

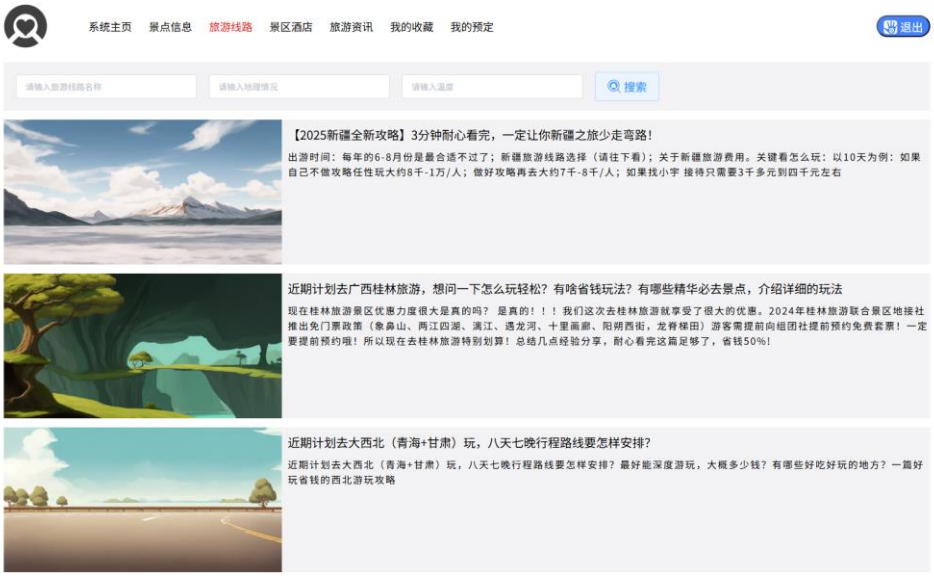

图 4.1.7 用户旅游线路列表界面

点击线路之后进入线路详情页面，展示图片、名称和内容，点击收藏按钮可以收藏线路，线路详情功能设计如下图所示：

图 4.1.8 旅游线路详情界面

点击导航栏的景区酒店进入到酒店列表，可以根据酒店、景点进行筛选，酒店列表功能设计如下图所示：

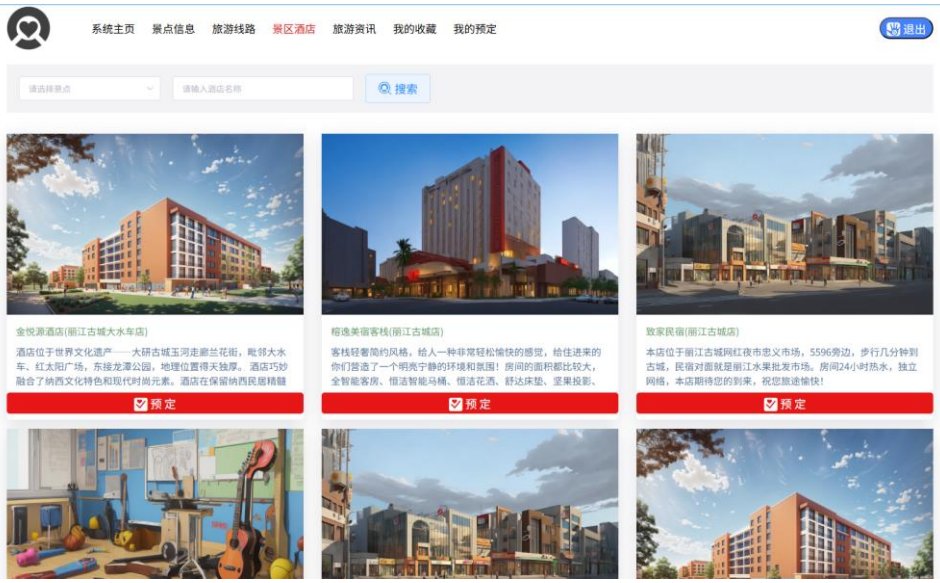

图 4.1.9 酒店列表中心界面

点击相应酒店进入到酒店详情页，酒店详情功能设计如下图所示：

图 4.1.10 酒店详情界面

点击导航栏的旅游资讯进入资讯列表，可以根据资讯名称筛选，资讯列表功能设计如下图所示：

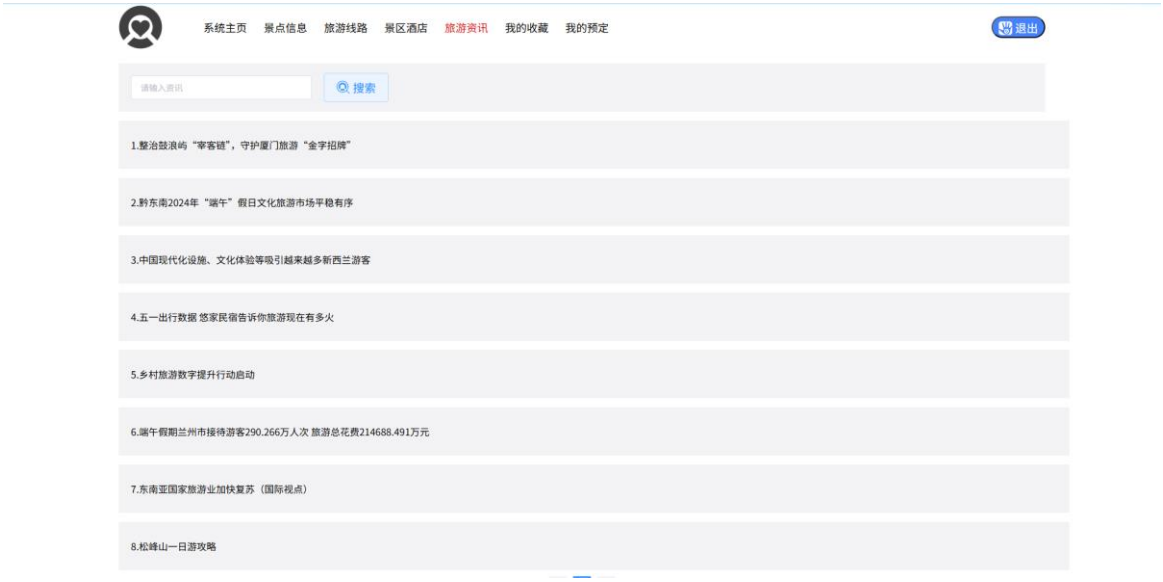

图 4.1.11 资讯列表界面

点击对应资讯进入资讯详情页，展示资讯的标题和具体内容，资讯详情功能设计如下图所示：

图 4.1.12 资讯详情界面

点击导航栏的我的收藏进入到收藏列表，点击收藏查看线路内容，收藏列表功能设计如下图所示：

图 4.1.13 我的收藏列表界面

点击导航栏的我的预定进入到预定列表，预定列表功能设计如下图所示：

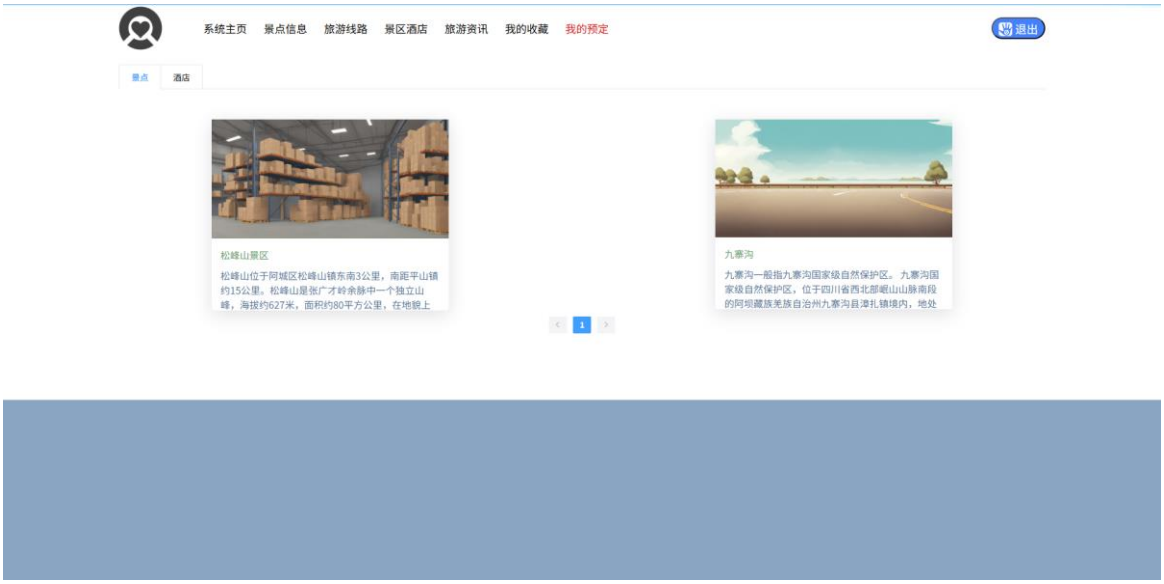

图 4.1.14 用户预定列表界面

点击导航栏头像，可以修改个人信息、修改密码和头像，个人中心功能设计如下图所示：

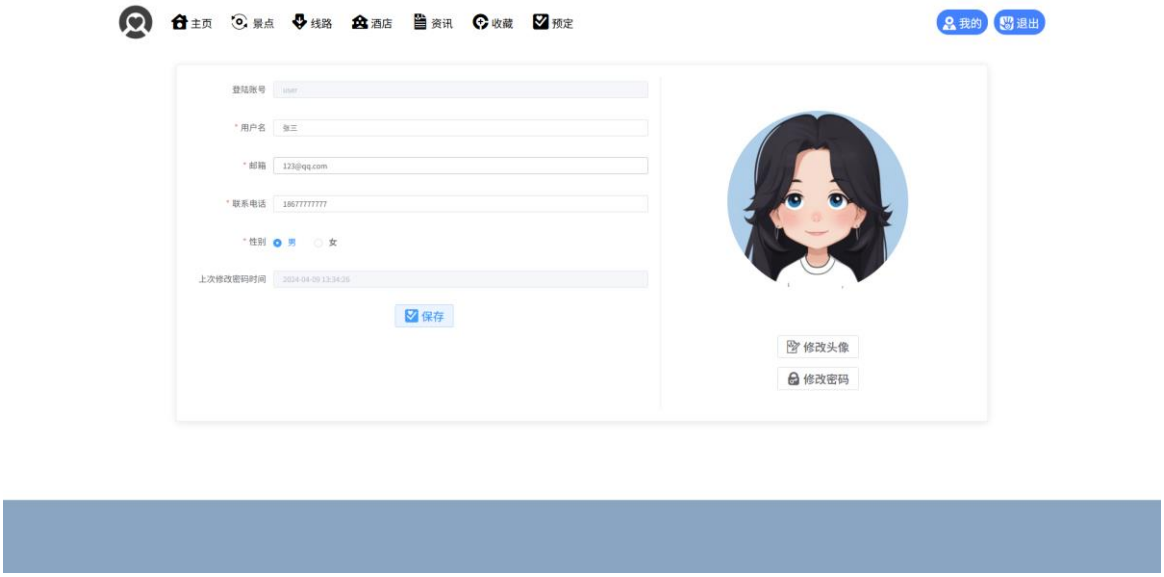

图 4.1.15 用户个人中心界面

4.2 管理员功能

管理员登陆后进入管理员首页，展示出平台的预约数据。功能如下图所示：

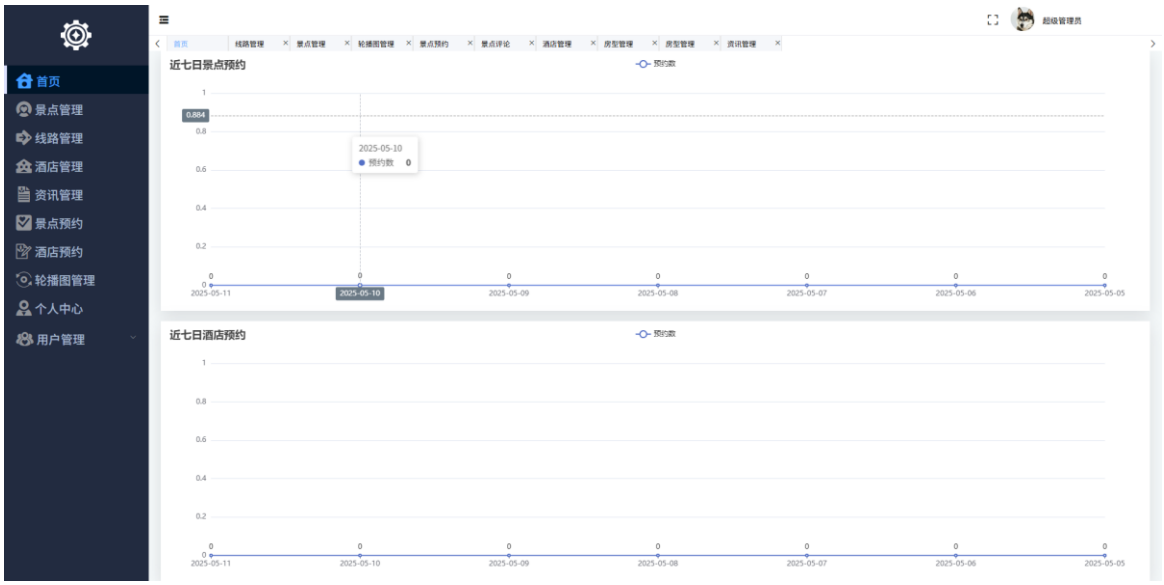

图 4.2.1 后台首页界面

点击景点管理，可以对该景点进行增加、删除、修改、查询，景点管理功能设计如下图所示：

图 4.2.2 景点管理界面

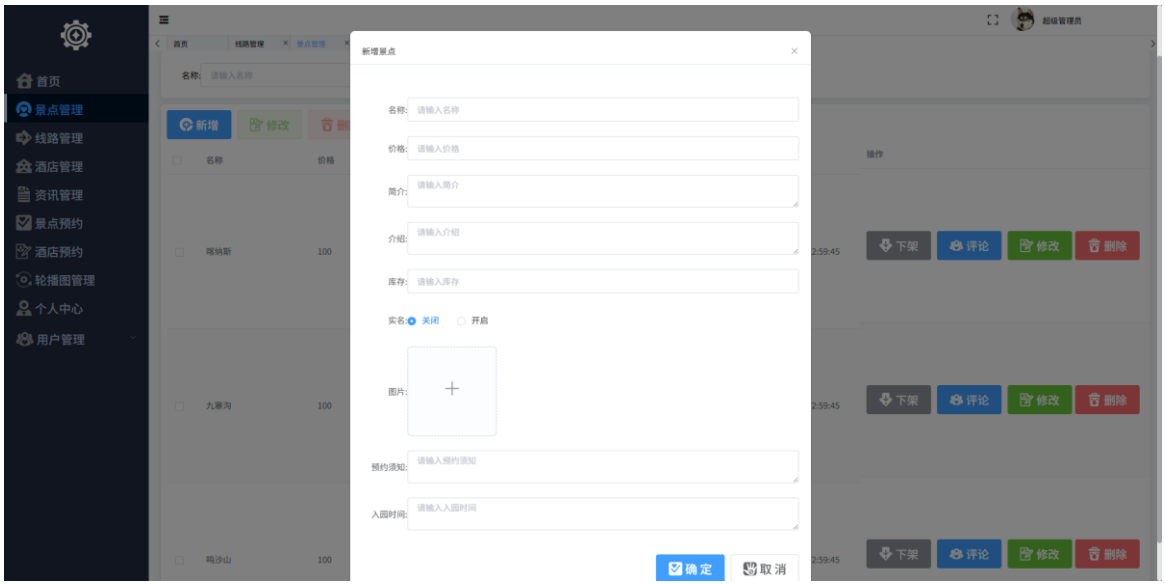

图 4.2.3 景点新增界面

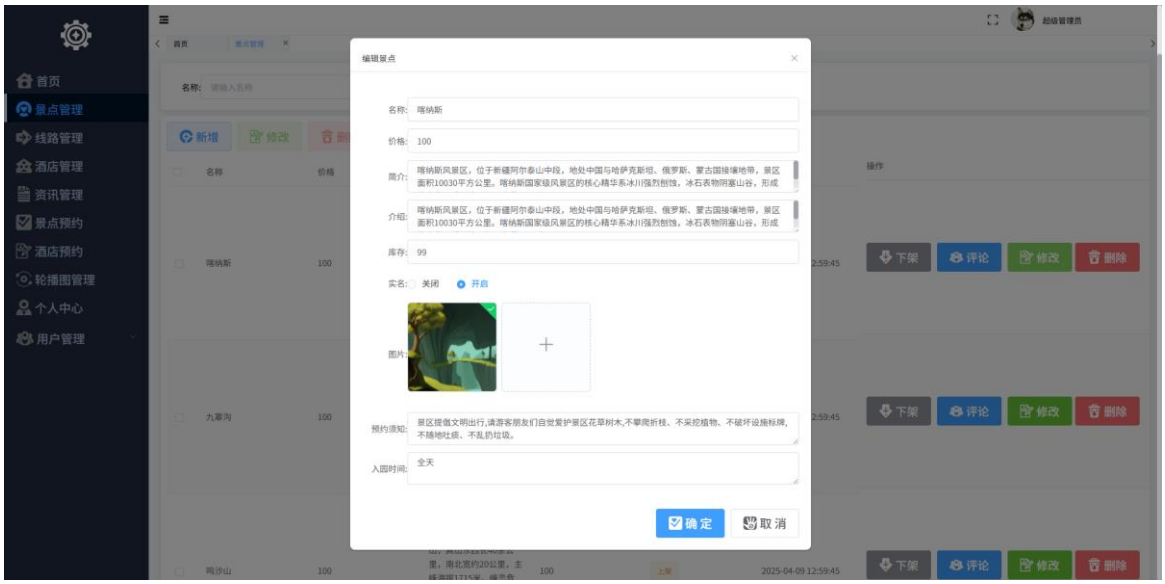

图 4.2.4 景点修改界面

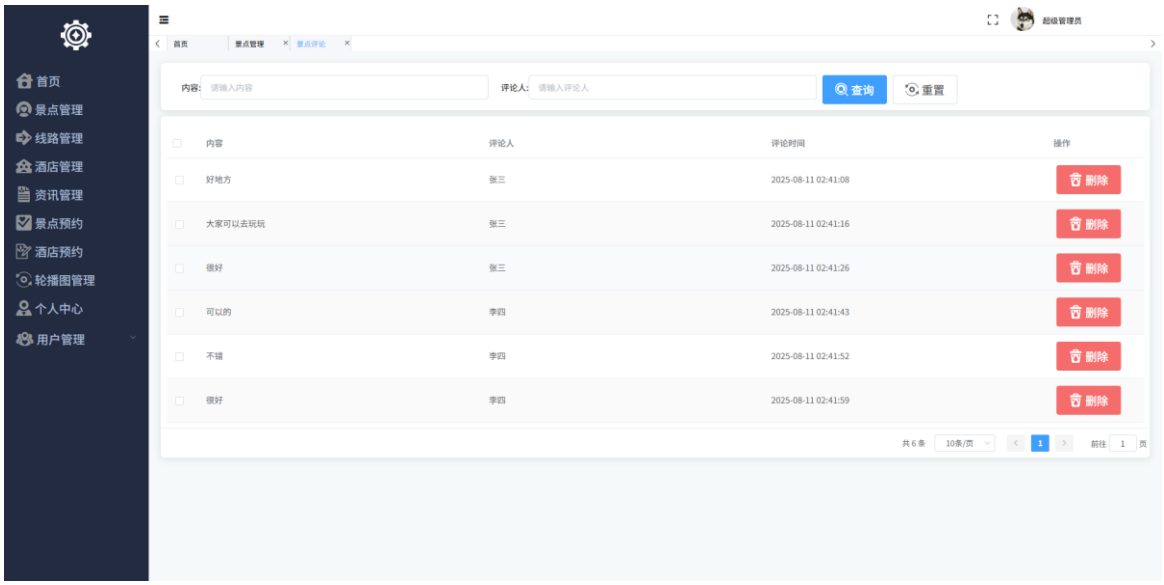

图 4.2.5 景点评论界面

点击线路管理，可以对线路进行增加、删除、修改、查询。线路管理功能设计如下图所示：

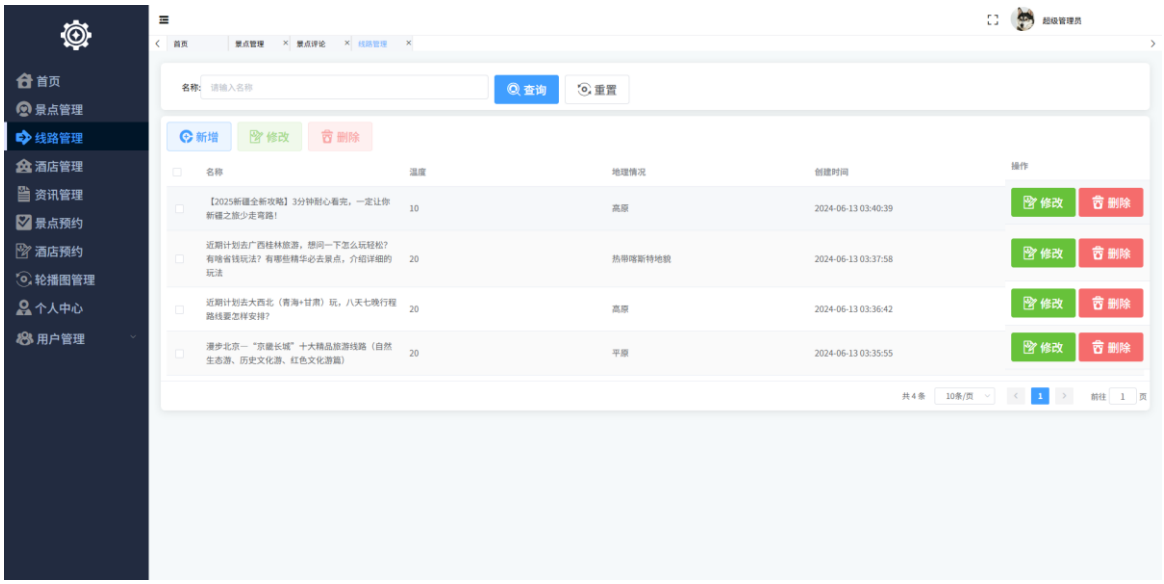

图 4.2.6 线路管理界面

点击酒店管理，可以新增、删除、修改、查询酒店，点击房型按钮可以对酒店的房型进行修改。酒店管理设计如下图所示：

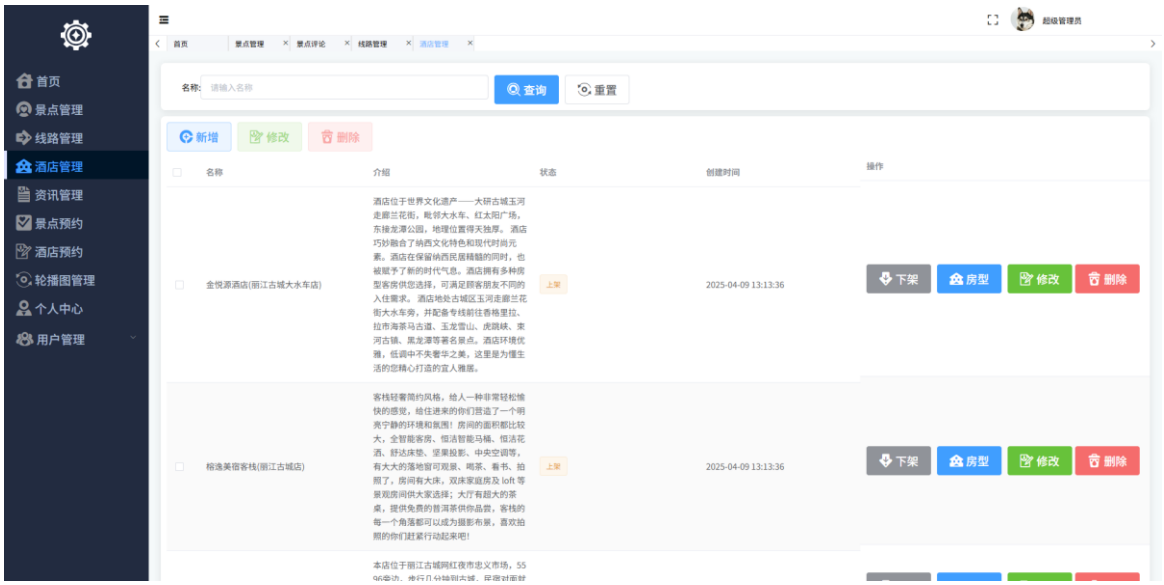

图 4.2.7 酒店管理界面

图 4.2.8 酒店新增界面

图 4.2.9 酒店修改界面

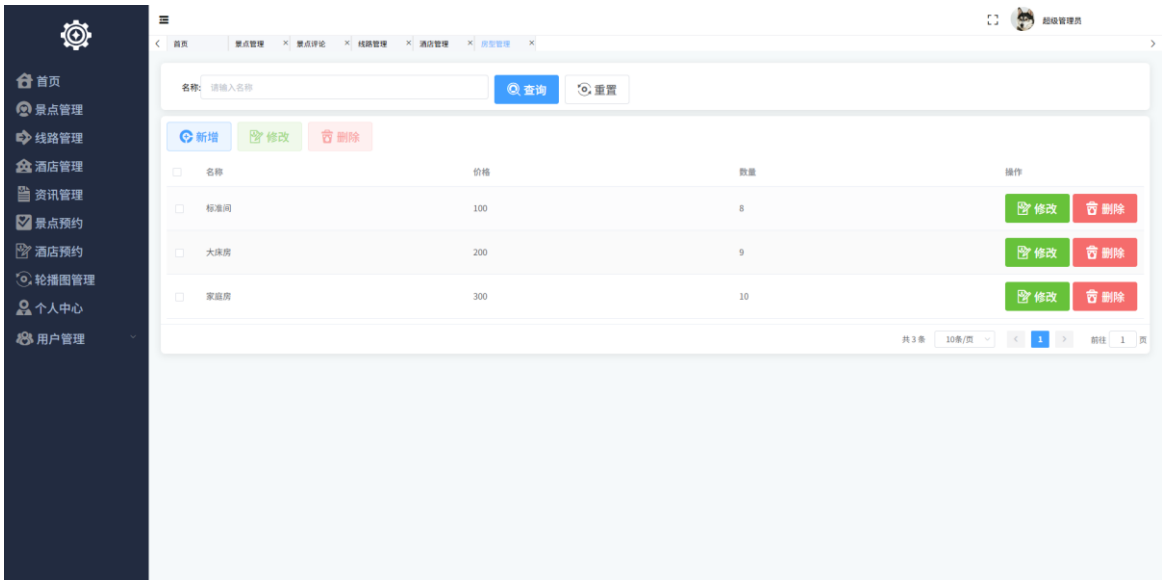

图 4.2.10 酒店房型界面

点击资讯管理，可以对资讯进行增加、删除、修改、查询。资讯管理如下图所示：

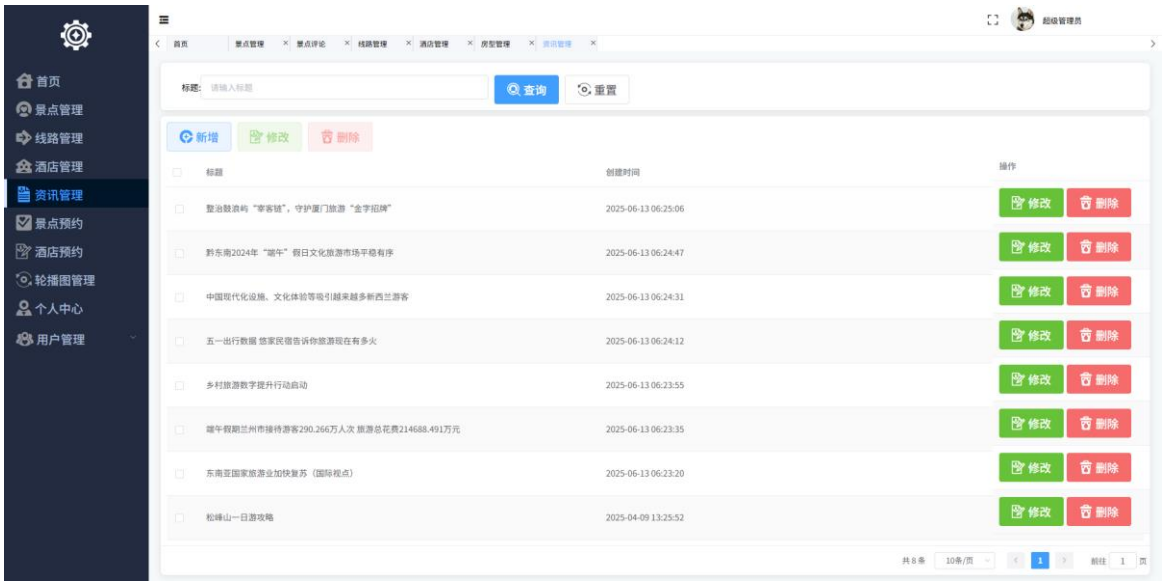

图 4.2.11 资讯管理界面

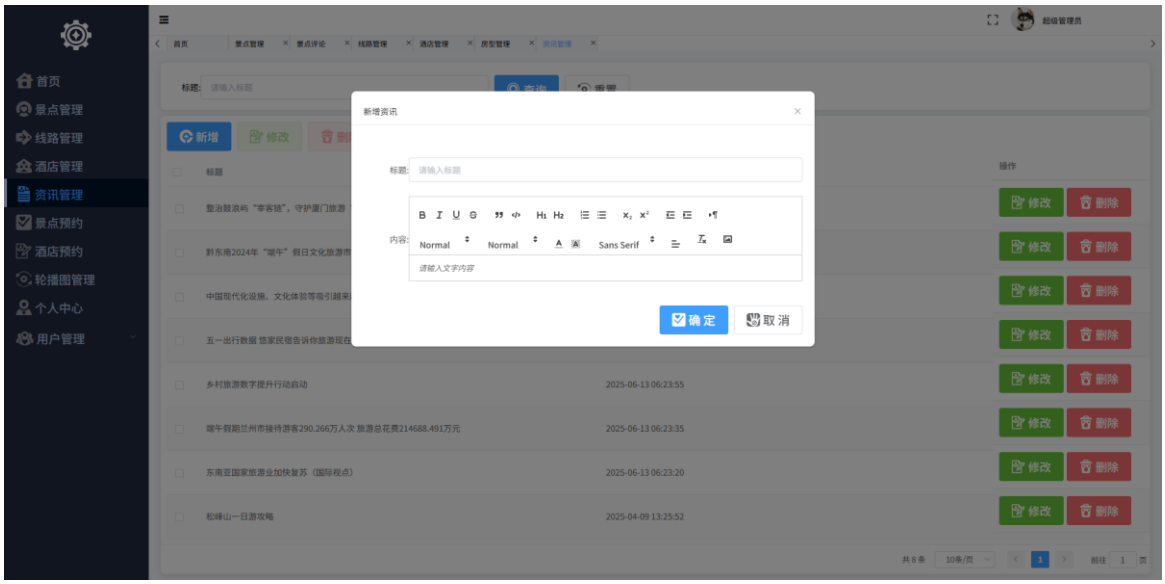

图 4.2.12 资讯新增界面

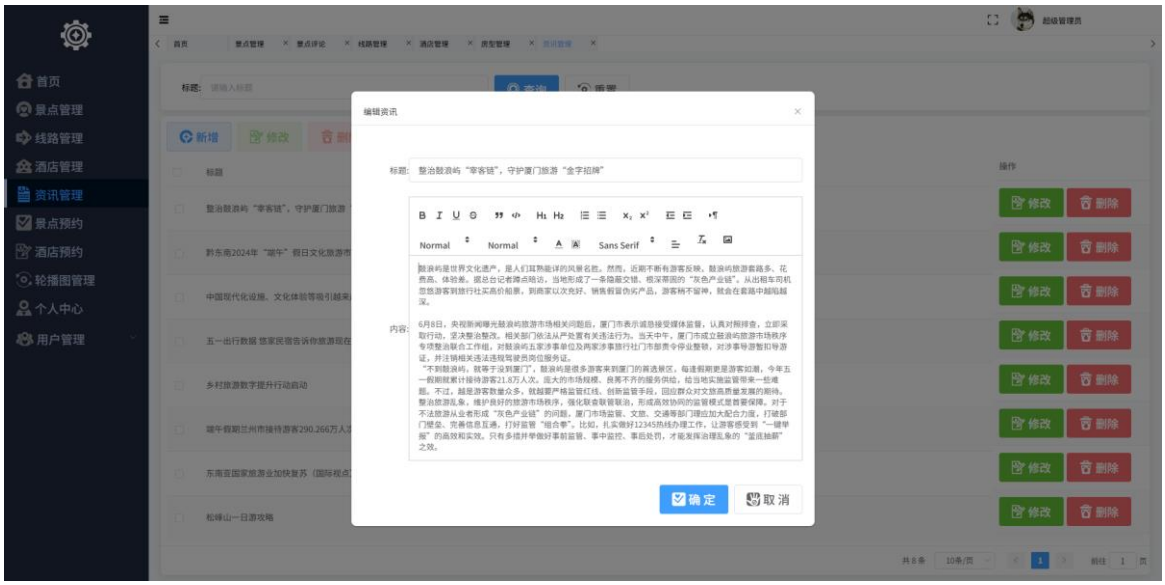

图 4.2.13 资讯修改界面

点击景点预约管理，可以对预约的订单进行删除、查询。还可以进行核销操作。景点预约管理如下图所示：

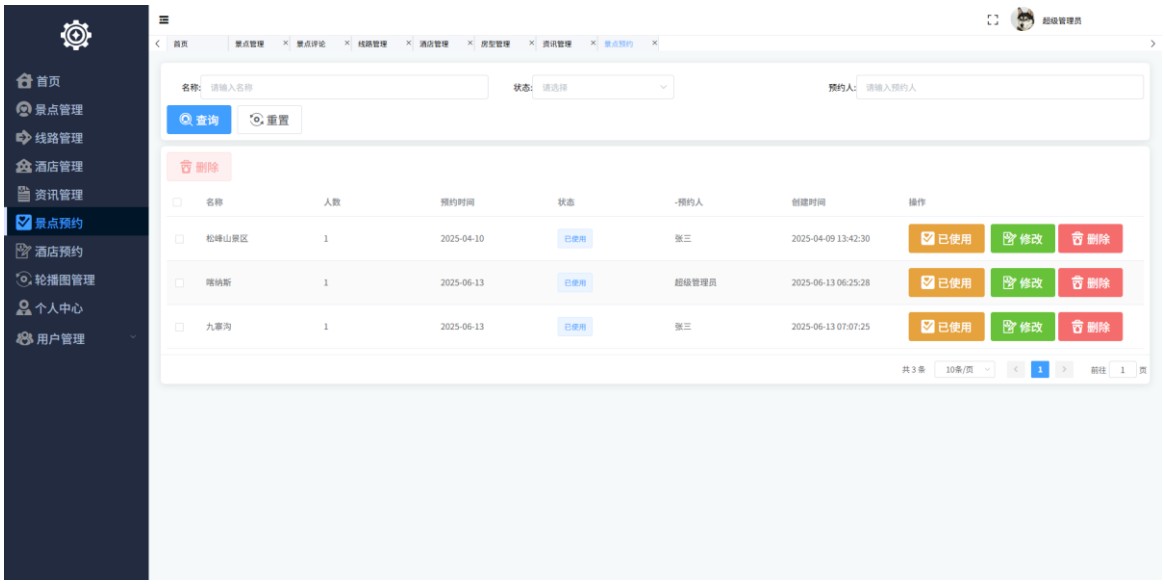

图 4.2.14 景点预约界面

点击酒店预约管理，可以对预约的订单进行删除、查询。还可以进行核销操作。酒店预约管理如下图所示：

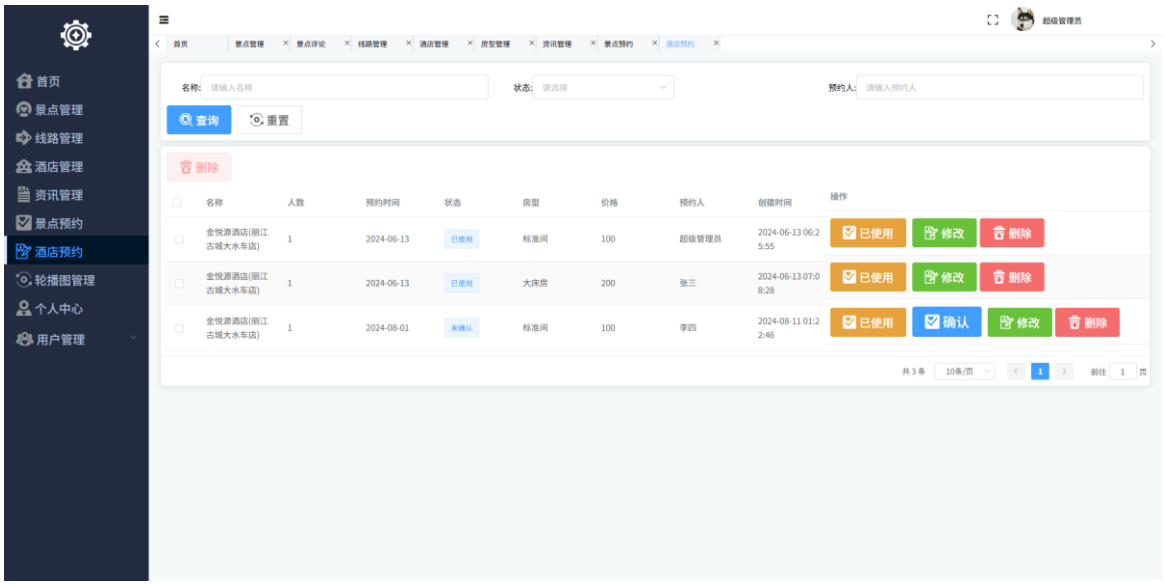

图 4.2.15 酒店预约界面

点击轮播图管理，可以对首页的轮播图进行删除、查询。轮播图管理如下图所示：

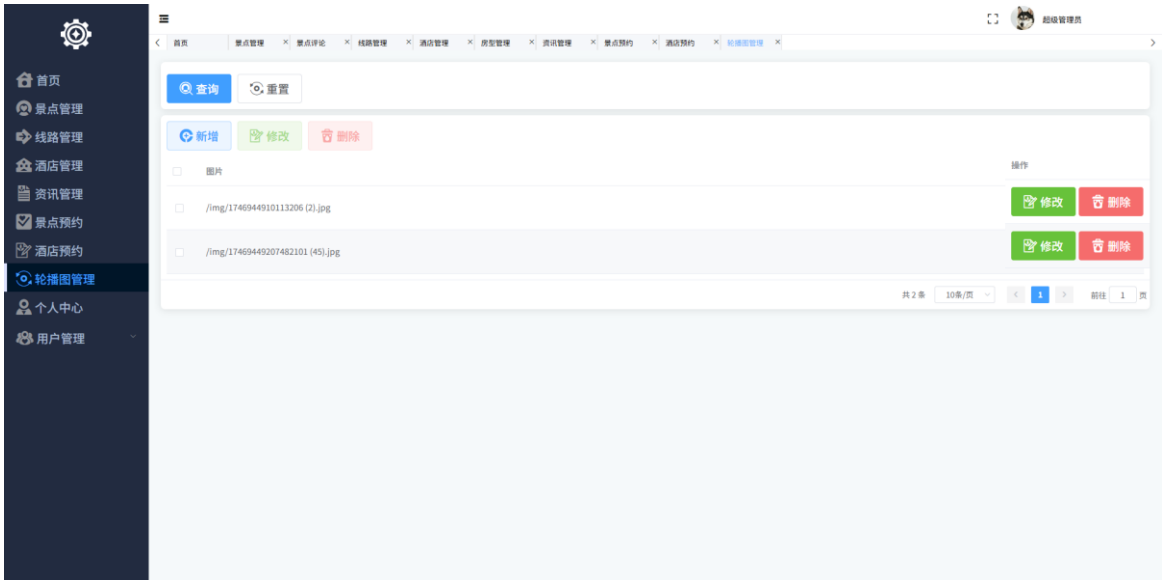

图 4.2.16 轮播图界面

点击个人中心，可以修改信息，修改密码和头像，个人中心功能设计如下图所示：

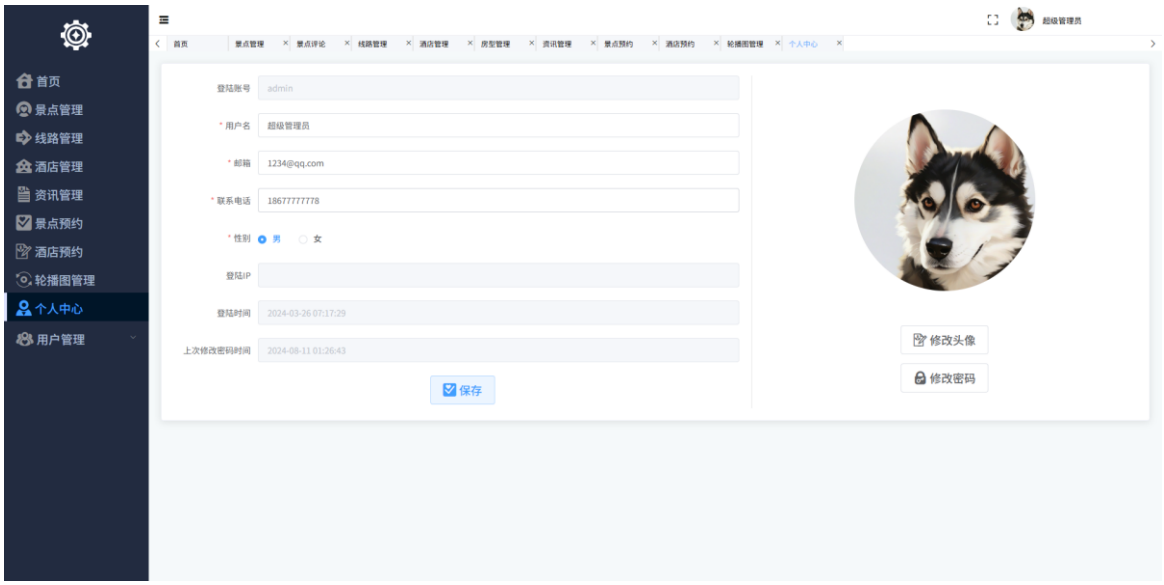

图 4.2.17 个人中心界面

第五章 系统测试

5.1 系统测试的目的

系统测试是软件开发生命周期中至关重要的一个环节。在这个阶段，开发团队需要将已经集成完毕的软件系统作为一个完整的计算机系统元素，与硬件环境、支撑软件平台、实际业务数据以及相关操作人员等系统要素进行有机结合，在模拟真实运行环境或实际生产环境中进行全面的测试验证。

系统测试的主要目的是通过执行一系列精心设计的测试用例，将软件系统的实际表现与预先定义的系统需求规格说明书进行详细比对，从而发现系统功能、性能等方面存在的任何不符合项或矛盾点。这种验证过程需要覆盖系统的所有功能模块，包括但不限于用户界面、业务逻辑、数据处理、系统接口等各个方面。测试人员需要特别关注系统在边界条件、异常情况和压力负载下的表现，确保系统在各种场景下都能稳定可靠地运行。

在具体实施过程中，系统测试通常包括功能测试、性能测试、安全性测试、兼容性测试等多个维度。功能测试主要验证系统是否按照需求规格正确实现了各项功能；性能测试则关注系统的响应速度、吞吐量、资源占用率等指标；安全性测试检查系统是否存在潜在的安全漏洞；兼容性测试确保系统能够在不同的硬件配置、操

作系统和浏览器环境下正常工作。通过这样全方位的测试验证，可以最大程度地确保软件系统满足业务需求和技术规范，为后续的上线部署奠定坚实基础。

5.2 系统测试分析

用户层：围绕用户界面的规范性、友好性、可操作性、系统对用户的支持，以及数据的安全性等方面展开。另外，用户层的测试通常还应注意可维护性测试和安全性测试。

应用层：主要是针对产品工程应用或行业应用的测试。从应用软件系统的角度出发，模拟实际应用环境，对系统的兼容性、可靠性等进行测试。针对整个系统的应用层测试，包含并发性能测试、负载测试、压力测试、强度测试、破坏性测试。

功能层：检测系统是否已经实现需求规格说明中定义的功能，以及系统功能之间是否存在类似共享资源访问冲突的情况。

子系统层：针对产品内部结构性能的测试。

协议/指标层：针对系统所支持的协议，进行协议一致性测试和协议互通测试。

系统测试是一个多层次、多维度的综合验证过程，需要从不同层级进行全面检测。在用户层测试中，除了界面规范性和操作友好性外，还需要重点关注用户体验的流畅度、界面元素的响应速度以及错误提示的明确性。可维护性测试要验证系统日志记录的完整性、错误定位的准确性；安全性测试则需覆盖用户权限管理、数据加密传输、防注入攻击等关键方面。

应用层测试需要模拟真实业务场景，不仅要测试系统在常规负载下的表现，更要关注峰值时期的稳定性。其中并发性能测试要模拟多用户同时操作系统的情况；负载测试需验证系统在持续高负载下的资源占用情况；压力测试要突破系统极限，找出性能瓶颈；破坏性测试则要模拟突发异常情况，检验系统的容错能力。这些测试都需要结合具体行业特点设计测试用例，如金融系统要特别关注交易一致性，电商系统则要重点测试促销时的高并发处理能力。

功能层测试要建立完整的测试用例库，覆盖所有业务场景，包括正常流程、异常流程和边界条件。特别要注意功能间的耦合关系，避免出现资源竞争或数据不一致的情况。子系统层测试要深入到模块内部，验证数据处理效率、算法准确性等核心指标。协议/指标层测试要确保系统与其他组件或第三方服务的交互符合标准规范，

保证系统的开放性和扩展性。通过这样层层递进的测试策略，才能全面保障软件系统的质量和可靠性。

5.3 系统测试的方法

功能测试：功能测试属于黑盒测试，是系统测试中最基本的测试。功能测试主要根据产品的需求规格说明和测试需求列表，验证产品是否符合需求规格说明。

在线学习系统

协议一致性测试：主要用于分布式系统。在分布式系统中，很多功能的实现是通过多台计算机相互协作来完成的，这要求计算机之间能相互交换信息，所以需要制定一些规则（协议）。对协议进行测试，通常包括：协议一致性测试、协议性能测试、协议互操作性测试、协议健壮性测试。

性能测试：主要用于实时系统和嵌入式系统，性能测试是指测试软件在集成系统中的运行性能，目标是量度系统的性能和预先定义的目标有多大差距。一种典型的性能测试是压力测试，当系统同时接收极大数量的用户和用户请求时，需要测量系统的应对能力。性能测试要有工具的支持，在某种情况下，测试人员必须自己开发专门的接口工具。

压力测试：又称强度测试，是在各种超负荷的情况下观察系统的运行情况的测试。

容量测试：在系统正常运行的范围内测试并确定系统能够处理的数据容量。容量测试是面向数据的，主要目的就是检测系统可以处理目标内确定的数据容量。

安全性测试：安全性测试就是要验证系统的保护机制是否抵御入侵者的攻击。保护测试是安全性测试中一种常见的测试，主要用于测试系统的信息保护机制。评价安全机制的性能与安全功能本身一样重要，其中安全性的性能主要包括：有效性、生存性、精确性、反应时间、吞吐量。

失效恢复测试：验证系统从软件或者硬件失效中恢复的能力。失效恢复测试采用各种人为干预方式使软件出错，造成人为的系统失效，进而检测系统的恢复能力。如果恢复需要人为干预，则应考虑平均修复时间是否在限定的范围内。

备份测试：备份测试是失效恢复测试的补充，目的是验证系统在软件或者硬件

失效的实践中备份其数据的能力。

GUI 测试：GUI 测试与用户友好性测试和可操作性测试有重复，但 GUI 测试更关注对图形界面的测试。GUI 测试分为两个部分，一方面是界面实现与界面设计的情况要符合；另一方面是要确认界面能够正确处理事件。GUI 测试设计测试用例一般要从以下 4 方面考虑：（1）划分界面元素，并根据界面的复杂性进行分层。通常把界面划分为三个层次，第一层是界面原子层；第二层是界面组合元素层；第三层是一个完整的窗口。（2）在不同的界面层次确定不同的测试策略。（3）进行测试数据分析，提取测试用例。（4）使用自动化测试工具进行脚本化工作。

健壮性测试：又称容错测试，用于测试系统在出故障时，是否能够自动恢复或者忽略故障继续运行。健壮性测试的一般方法是软件故障插入测试，在软件故障插入测试中，需要关注三个方面：目标系统、故障类型和插入故障的方法。

兼容性测试：检验被测的应用系统对其他系统的兼容性。

易用性测试：与可操作性类似。检测用户在理解和使用系统方面是否方便。易用性测试是面向用户的系统测试，包括对被测系统的系统功能、系统发布、帮助文本和过程等的测试。

总结

本系统还存在一些不完善的地方，业务逻辑方面编写效率不够高，界面设计色彩相对比较单调，用了一个静态模板，页面的布局与设计水平有待提高，界面过于简单不符合审美，逻辑可能有疏忽的地方这些都是后期需要改进的。我们会不断的学习新的知识去完善我们的系统。

本系统服务器端使用 SpringBoot 框架，该框架大大简化了整个项目的开发，减少了繁杂冗余的配置文件，改用统一的 yml 文件，文件格式要求严格，但是内容一目了然，通过数据库的表，一键生成你需要的 dao 层，service 层，controller 层，大大简化开发时间，不用再把时间浪费在重复简单的代码编写上，把时间和精力放到核心的业务逻辑处理上，同时改该框架可以很方便的对一些增删改查请求做出响应。经历了前期需求分析，概要设计，详细设计等一系列步骤，本系统逐步走向完善，最终得以完成本设计，前端使用 Vue 和 Element 框架实现前端的设计，简化代码的开发，便于功能的实现。