



闽江学院

本科毕业论文（设计）

题 目	基于 SpringBoot 的三坊七巷景区旅游系统 的设计与实现
学 生 姓 名	吴建煌
学 号	3187103216
学 院	数学与数据科学学院（软件学院）
年 级	2018 级
专 业	软件工程
指 导 教 师	曹永忠
职 称	助理教授
完 成 日 期	2022 年 4 月 29 日

闽江学院毕业论文（设计）诚信声明书

本人郑重声明：

兹提交的毕业论文（设计）《基于 SpringBoot 的三坊七巷景区旅游系统的设计与实现》，是本人在指导老师 曹永忠 的指导下独立研究、撰写的成果；论文（设计）未剽窃、抄袭他人的学术观点、思想和成果，未篡改研究数据，论文（设计）中所引用的文字、研究成果均已在论文（设计）中以明确的方式标明；在毕业论文（设计）工作过程中，本人恪守学术规范，遵守学校有关规定，依法享有和承担由此论文（设计）产生的权利和责任。

声明人（签名）：

年 月 日

摘 要

改革开放以来，百姓生活也越来越好，人们对消遣娱乐的需求也越来越大，随之出现了越来越多的旅游景区供人们选择。近些年，几乎每个人都拥有计算机设备，越来越多的东西都变成线上化，对于三坊七巷景区来说也不例外。用户通过网站即可完成对三坊七巷景区的了解，且网站交互逻辑简单，无需学习成本，给游客带来了更加便利的服务。

系统采用 B/S 架构进行开发，相对于传统的 C/S 架构，B/S 架构更利于开发、维护和升级，可以实现跨平台，只需在有操作系统和浏览器的设备即可访问。系统的设计采用了数据持久层、业务逻辑层和页面展示层三层架构设计，使用当今最主流的 SpringBoot 框架开发，MySQL 作为数据库，让开发过程变得更加规范化。同时系统的界面设计友好，操作合理，具备了较强的适用性。

关键词：旅游系统；三坊七巷；SpringBoot；MySQL

Abstract

Since the reform and opening up, people's life has become better and better, and people's demand for entertainment has become greater and greater. With it, there are more and more scenic spots for people to choose. In recent years, almost everyone has computer equipment, and more and more things have become online, including Sanfang and Qixiang scenic spots. Through the website, users can understand the scenic spots of Sanfang and Qixiang, and the interactive logic of the website is simple without learning cost, which brings more convenient services to tourists.

The system adopts B / S architecture for development. Compared with the traditional C / S architecture, B / S architecture is more conducive to development, maintenance and upgrading. It can realize cross platform and can be accessed only on the device with operating system and browser. The design of the system adopts the three-tier architecture design of data persistence layer, business logic layer and page display layer. It is developed with the most mainstream springboot framework and MySQL as the database, which makes the development process more standardized. At the same time, the system has friendly interface design, reasonable operation and strong applicability.

Key words: Tourism system; Three lanes and seven lanes; SpringBoot; MySQL

目 录

1 绪论	1
1.1 课题背景	1
1.2 研究的目的和意义	1
1.3 研究的内容	1
1.4 国内外研究状况	2
2 系统开发关键技术介绍	3
2.1 开发语言	3
2.2 开发工具	3
2.3 服务器技术	3
2.4 数据库技术	3
3 需求分析	4
3.1 可行性分析	4
3.1.1 社会可行性	4
3.1.2 技术可行性	4
3.1.3 操作可行性	4
3.2 业务需求分析	4
3.3 系统整体功能需求分析	4
3.3.1 用户模块的功能需求	4
3.3.2 管理员模块的功能需求	6
3.3.3 酒店人员模块的功能需求	7
3.4 非功能性需求分析	8
3.5 安全性需求分析	8
4 系统概要设计	9
4.1 系统设计思想	9
4.1.1 前台用户模块	9

4.1.2 后台管理模块	9
4.1.3 后台酒店人员模块	9
4.2 总体功能结构设计	10
4.3 数据库设计	10
4.3.1 数据库环境	10
4.3.2 数据关系模型	11
4.3.3 数据库表结构设计	12
5 系统详细设计与实现	19
5.1 前台用户系统功能模块设计与实现	19
5.1.1 用户注册	19
5.1.2 用户登录	21
5.1.3 个人信息编辑	22
5.1.4 预订酒店	23
5.1.5 预订景点	26
5.1.6 我的预订	27
5.1.7 发布攻略	29
5.1.8 关注攻略	30
5.1.9 关注路线	31
5.2 后台管理系统功能模块设计与实现	32
5.2.1 后台首页模块	32
5.2.2 用户管理模块	33
5.2.3 酒店管理模块	35
5.2.4 景点管理模块	37
5.2.5 攻略管理模块	39
5.2.6 路线管理模块	41
5.3 后台酒店管理员系统功能模块设计与实现	43
5.3.1 酒店管理员系统登录	43
5.3.2 酒店订单审核	44
5.3.3 酒店价格设置	45

6 系统测试	46
6.1 测试目的	46
6.2 系统的功能测试	46
6.2.1 前台用户系统功能测试	46
6.2.2 后台管理员系统功能测试	48
6.2.3 后台酒店人员系统功能测试	50
6.3 测试总结	51
7 总结与展望	52
参考文献.....	53
致谢.....	54

基于 SpringBoot 的三坊七巷景区旅游系统的设计与实现

吴建煌

(闽江学院 数学与数据科学学院 (软件学院), 福建 福州 350108)

1 绪论

1.1 课题背景

近年来,各旅游景点的热度不断攀升,许多著名的旅游景点也有相应的旅游网站来为游客提供更好的旅游体验。三坊七巷作为我国第一批历史文化街区,是福州城市历史发展的标志^{[1][2]}。因此,对于三坊七巷景区和游客来说,一个稳定好用的旅游系统是非常有必要的。

1.2 研究的目的和意义

现阶段,旅游景区处于井喷式发展,游客对游玩的体验要求也越来越高,传统的景区运营方式费时又费力。在互联网技术高度普及的今天,Web 服务已经成为了一种主流的技术,人们使用具有联网的设备即可访问系统^[3]。

旅游系统的建设是目前旅游市场的趋势,可以通过互联网解决传统旅游业无法解决的需求。同时,旅游系统涉及到旅游服务出行、住宿、景点等,可以大大提高业务的来源和服务水准。三坊七巷景区旅游系统可以对景区进行更好的宣传,同时能让更多的游客体验景区的产品特色。

1.3 研究的内容

系统区分了普通用户(游客)、管理员用户和酒店管理人员,不同用户的权限和功能需求也是不同的。

1.4 国内外研究状况

我国的景区旅游系统从二十世纪九十年代就开始发展，早起诞生了如华夏旅游网、艺龙旅游网等许多知名网站，后续随着单个景区的火爆，也诞生了许多针对一个景区的旅游网站。现如今，随着系统开发的成本不断降低，国内越来越多的景区愿意建设属于自己的系统。

与国内相比，国外的旅游系统起步较早，尤其是欧美发达国家，使用旅游系统的用户在网名占比中比我国会高一些。不管国内还是国外，旅游系统的开发对景区来说已经是一种趋势，都旨在为游客提供更好的服务。

2 系统开发关键技术介绍

2.1 开发语言

系统的后端采用了 Java 语言，SpringBoot、MyBatis 等框架协助开发。Java 语言具有健壮、安全、多线程等特点^[4]。Java 语言的平台无关性可以使系统实现平台无关，只需编译一次，就可以在不同平台运行使用^{[5][6]}。

系统前端使用 Thymeleaf 前端模板引擎，使用 html、css、js 前端三大组件开发，分别对应于网页的结构、样式和交互^[7]。应用了 bootstrap、layui、echarts 等响应式布局技术。

2.2 开发工具

系统开发过程中，主要使用的是 IntelliJ IDEA 集成开发工具，IDEA 是一个专注于 Java 语言的集成开发环境，整合了开发过程中实用的众多功能，可以加快开发速度^[8]。

2.3 服务器技术

系统开发过程中，在本机进行运行，因为使用的是 SpringBoot 技术，它内嵌了 Tomcat 服务器，Tomcat 是一个 servlet 容器，支持 Servlet 和 JSP 标准，部署简单且集成便利^[9]。

系统开发结束后，使用 maven 构建工具将项目打成 jar 包放在阿里云的 CentOS 中运行，Linux 有结构清晰、源代码开放性等方面优势，是部署环境的不二之选^[10]。

2.4 数据库技术

系统采用了 MySQL 关系型数据库，作为目前最热门的数据库技术，MySQL 数据库因其开放源码、运行速度快、磁盘空间占用少等优点，得到了广泛的应用^[11]。

3 需求分析

3.1 可行性分析

3.1.1 社会可行性

三坊七巷景区旅游系统的研发极大地方便了游客和景区从业人员，游客不再需要到处收集景区信息，还可以直接通过系统进行景点和酒店的预订，能给游客更好的游玩体验。

3.1.2 技术可行性

系统基于 B/S 架构开发，在 B/S 架构下，用户可以通过浏览器打开工作界面^[12]。采用 SpringBoot、MyBatis、MySQL、HTML、CSS、JavaScript 等技术，目前这些技术都是最主流的技术，网站在 Windows 系统中开发，部署于 Linux 中。

3.1.3 操作可行性

系统的前端使用响应式布局，可以实现一站通用，节约了时间成本和开发成本^[13]。页面设计简单，提示信息完整，易于操作，用户容易上手。

3.2 业务需求分析

三坊七巷景区旅游系统的整体目标是为游客带来更优质的服务和为景区带来更高效的管理方式，系统能够根据用户的操作数据来准确的反映景区酒店、景点等的收益情况，进而提高景区经济效益，也给用户提供更优质的服务。

3.3 系统整体功能需求分析

3.3.1 用户模块的功能需求

前台用户模块：首页热门酒店、景点图片展示，可以查看详情和根据条件进行搜索。具体功能需求有：

(1) 注册登录系统：用户输入用户名和密码，系统先确定其用户名的唯一性和密码的长度，若都符合则成功。

(2) 个人信息维护：用户名是区分各用户的唯一标识，具有唯一性，注册后就不能修改，可以完善自己的个人信息，如昵称、电话和邮箱等，也可对密码进行修改。

(3) 酒店信息查看：用户登录后可在首页中查看酒店的具体信息并进行预订操作，后台将会对预订的订单进行审核，用户可实时查看自己的订单状态。

(4) 景点信息浏览：可以浏览景点的介绍信息和预订收费景点，输入门票数量和联系方式后即可预订成功。

(5) 攻略信息查看：用户不需登录即可查看热门攻略，登录后可以收藏攻略和发布攻略。

(6) 路线信息浏览。

前台用户的功能用例图如 3-1 所示：

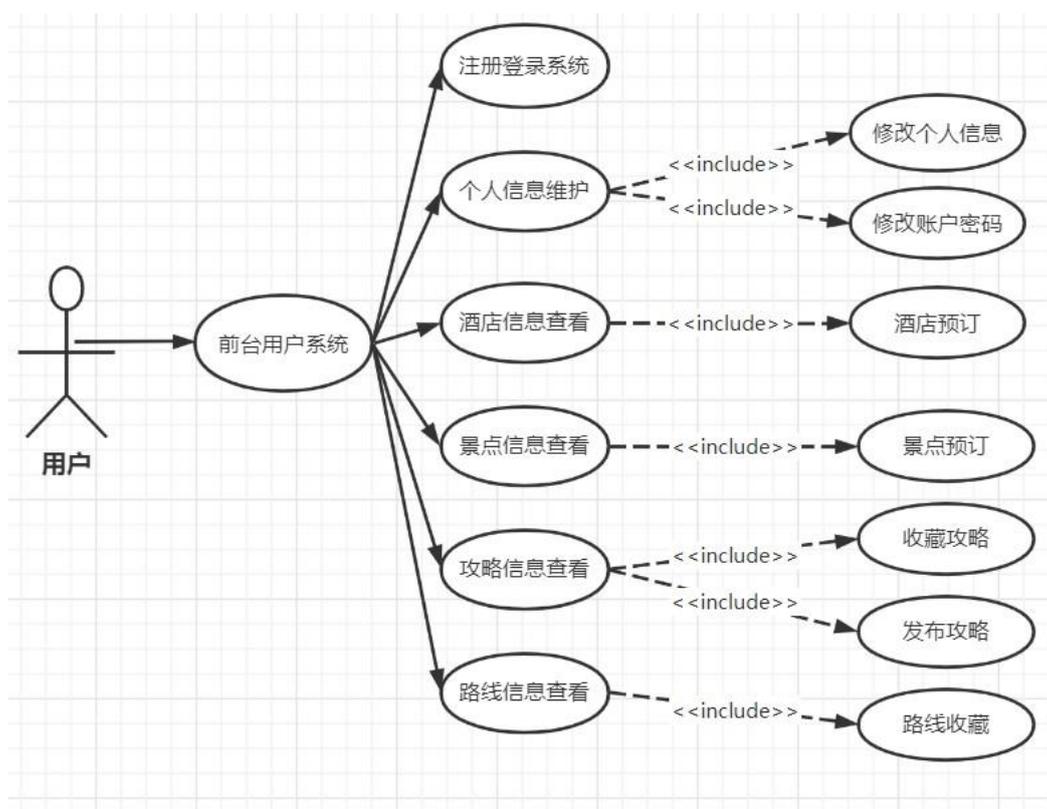


图 3-1 用户功能用例图

3.3.2 管理员模块的功能需求

后台管理模块：首页用户、酒店、景点等数据统计，管理员可以对系统的数据实现管理。具体的功能需求有：

(1) 登录系统：输入管理员账号后即可实时查看该账号是否有效，账号有效即可输入密码和验证码进行登录。

(2) 数据展示：首页可以对用户数、酒店数进行统计并以卡片形式展示，对常用的功能有快捷入口可以直达。

(3) 用户管理：可以查看系统所有的用户系统和对用户进行注销操作，该用户则无法登录系统。

(4) 酒店管理：对系统的所有酒店信息进行管理，设置不同酒店不同房型的价格，可以根据季节、人口旺季等实时修改酒店的价格。

(5) 景点管理：可以对景点进行添加，改变景点的信息，将景信息展示在用户系统首页中。

(6) 攻略管理：对攻略的信息进行管理和审核操作，审核通过的攻略可以展示在用户端的攻略信息中，不通过需要说明攻略审核失败的因素。

(7) 路线管理：可以查看到所有路线信息并进行编辑等操作，根据用户的收藏数进行数据统计，分析各个路线的受欢迎程度。

后台管理员的功能用例图如 3-2 所示：

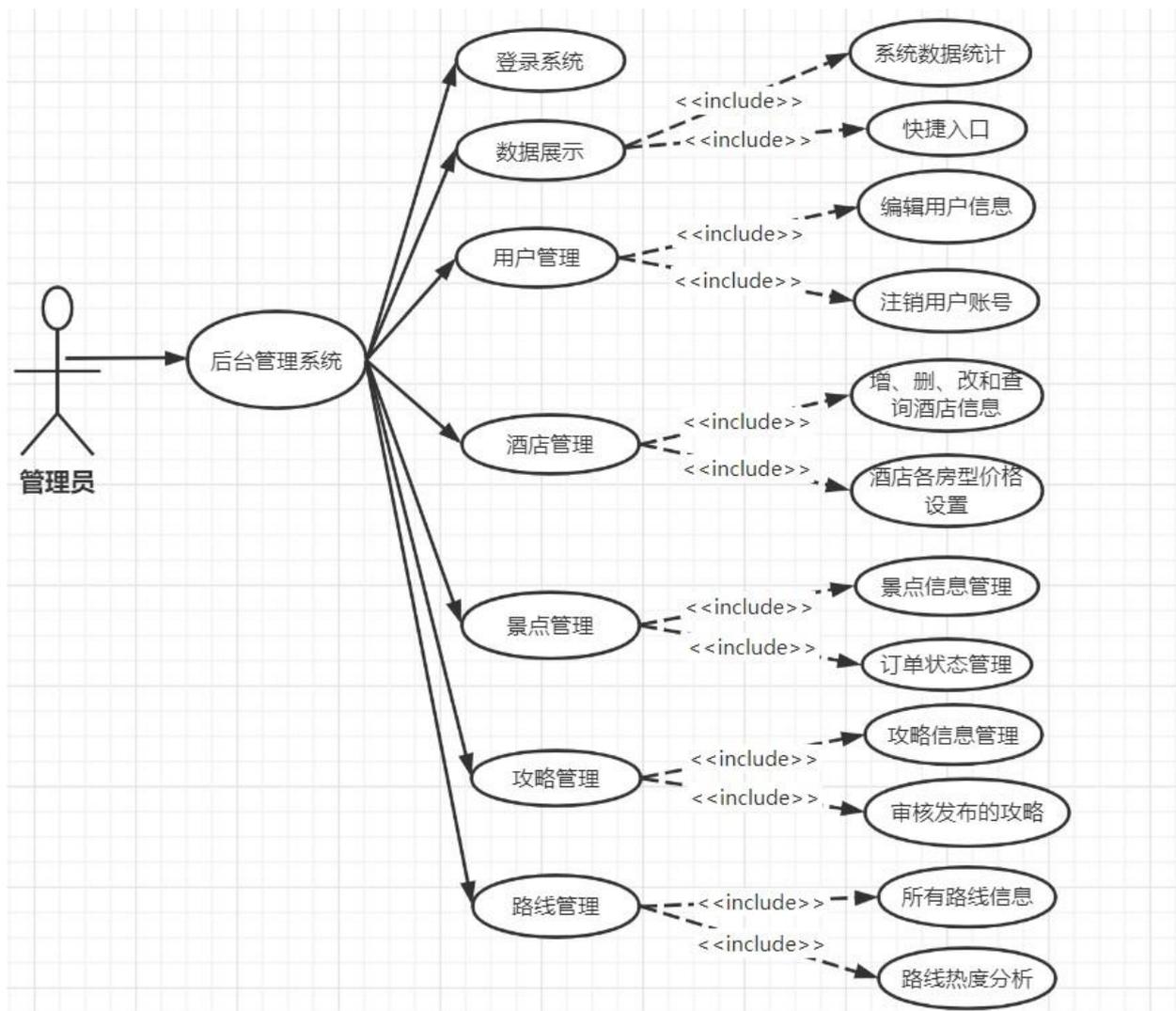


图 3-2 管理员功能用例图

3.3.3 酒店人员模块的功能需求

- (1) 登录系统：系统为每个酒店配备一个账号，登录系统后能对酒店进行一系列操作。
- (2) 审核酒店订单：酒店员工可以对自家酒店的订单进行状态审核操作。
- (3) 设置酒店价格：酒店人员可以根据酒店热度、季节等因素对自己酒店的各房型进行价格修改设置。

酒店人员功能用例图如 3-3 所示：

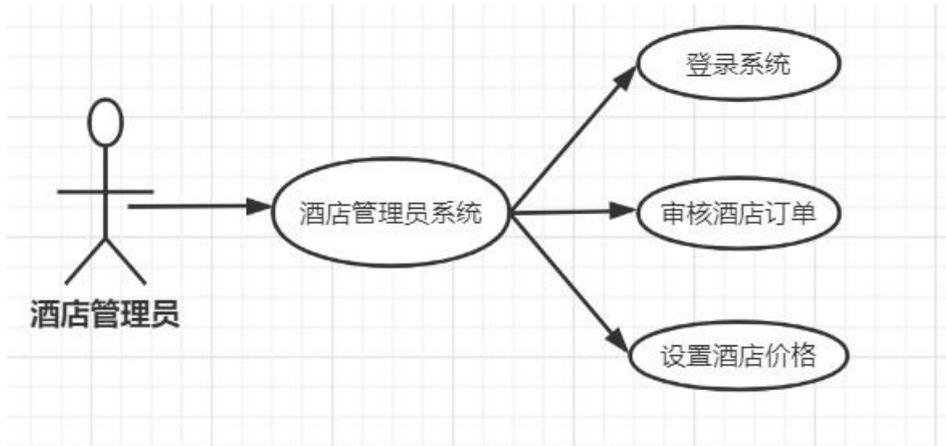


图 3-3 酒店人员功能用例图

3.4 非功能性需求分析

在需求分析时，非功能性需求也同样很重要，非功能性需求需要满足系统的稳定性、安全性、可移植性、可拓展性和可兼容性^[14]。系统的非功能性需求分析如下：

- (1) 性能方面：系统可以使用数据库缓存来减少调取数据库的次数，给数据库减轻压力，也可加快查询数据的速度，能给用户更好的使用体验。
- (2) 经济方面：系统的开发所使用的技术都是开源的且网上资料多，学习成本不会很高，在满足系统的功能前提下，尽可能的节约开发成本。
- (3) 扩展性：系统的开发不应当只专注于现在，更应该着眼于未来，系统需要升级接口，以便于日后的升级改造。
- (4) 兼容性：系统应该尽量做到多种操作系统、多种浏览器、多种客户端都能操作。

3.5 安全性需求分析

安全是系统的重要组成部分，系统应该保证数据的安全性，包括用户信息、企业信息等。可以对密码使用 MD5 加解密方式来保证密码的安全性，对数据库可以使用事务进行处理。

4 系统概要设计

4.1 系统设计思想

4.1.1 前台用户模块

游客能够在首页查看景区的相关酒店、景点、攻略和路线等详细信息，注册登录后可以根据自己的需求预订酒店和收费景点，收藏关注自己喜欢的旅游攻略和路线，游玩结束后也可以发布自己的旅游攻略，攻略审核通过后其他游客可以查看到。用户的在系统中的操作管理员都可以在后台管理系统中查看，对用户的数据进行分析，给用户提供更加优质的服务。

4.1.2 后台管理模块

后台管理系统是面向管理员的，管理员可以根据用户对酒店、景点的预订情况进行分析，可以查看到哪些酒店、景点预订的用户数多，并做出相应调整。管理员可以对用户发布的攻略进行审核操作，通过的攻略信息将会被显示在首页中供其他游客收藏关注，可查看用户关注的攻略、路线情况，并对关注情况进行热度分析。

4.1.3 后台酒店人员模块

系统针对各酒店管理者设计了一个酒店人员模块，酒店的工作人员需要使用管理员分配的酒店账号登录系统，进行审核自家酒店的用户预订订单和设置自家酒店的房型价格。

4.2 总体功能结构设计

三坊七巷景区旅游系统主要为游客提供了酒店和景点的预订，为景区工作人员提供了更便捷的管理方式。系统的功能模块图如图 4-1：

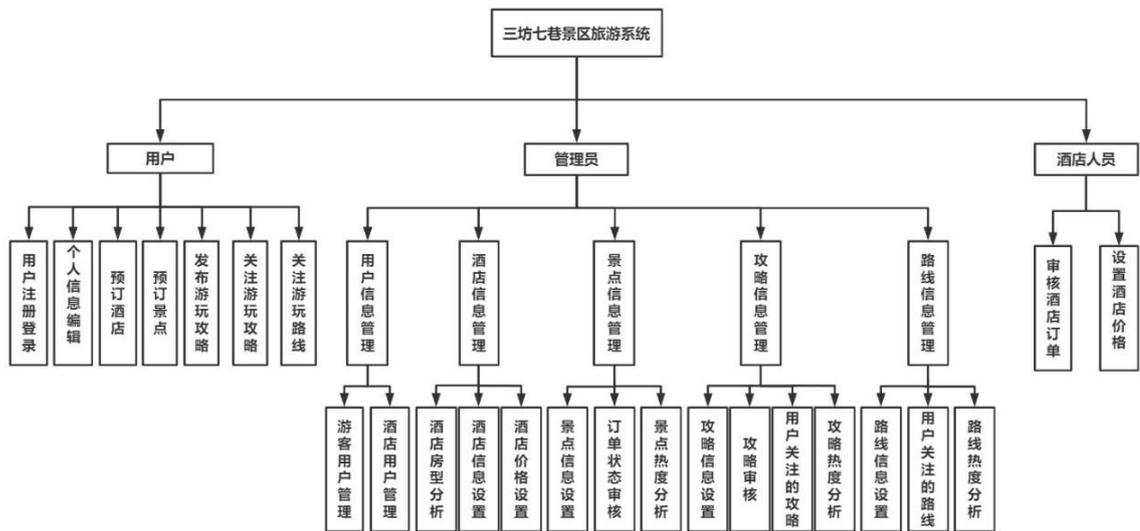


图 4-1 系统整体功能图

4.3 数据库设计

数据库作为系统的数据持久层，系统的数据信息存储都依赖于数据库，本系统采用了关系型数据库 MySQL。

4.3.1 数据库环境

数据库版本：MySQL 8.0

数据库的图形化界面工具：Navicat Premium 15

4.3.2 数据关系模型

系统整体的数据关系模型如图 4-2, 4-3 所示：

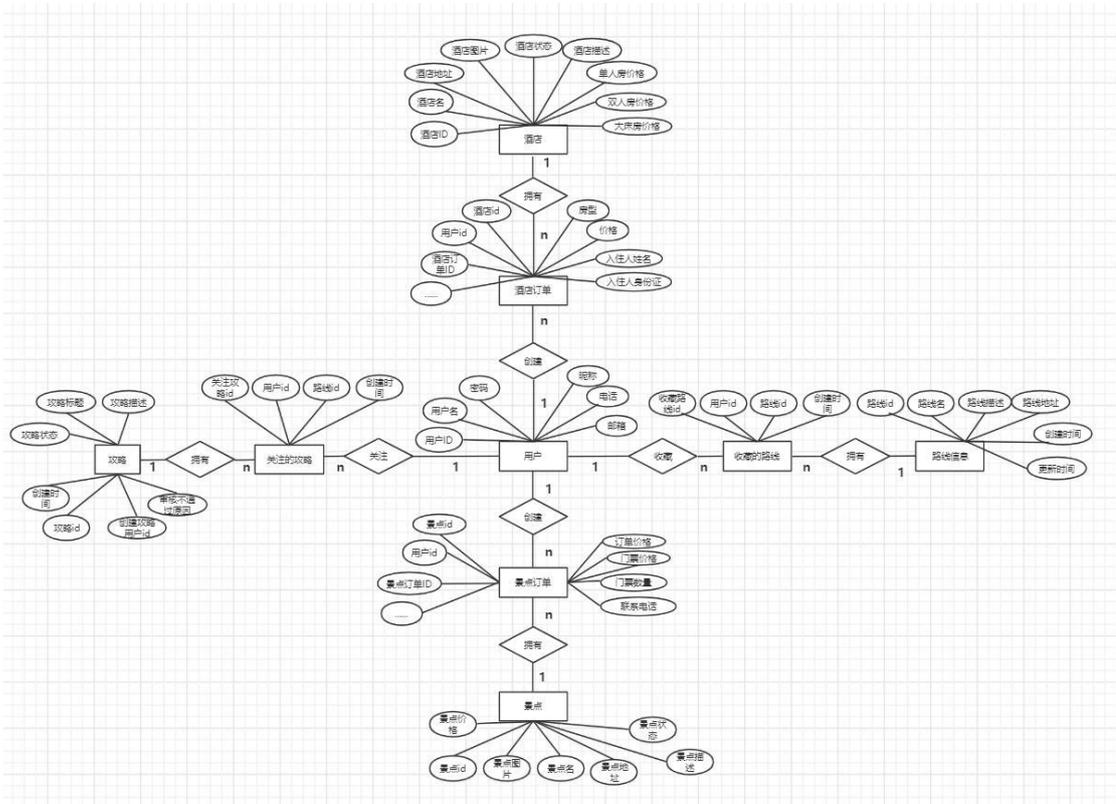


图 4-2 系统整体 E-R 图

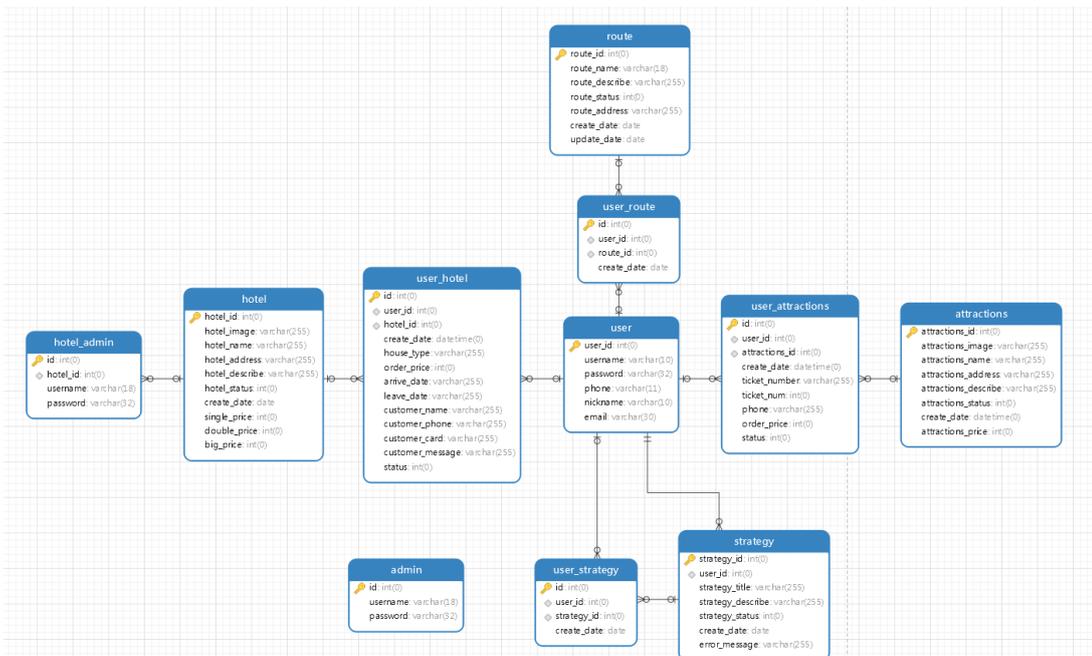


图 4-3 系统整体关系模型图

4.3.3 数据库表结构设计

(1) 用户需要进行账号的登录才能进入系统，需要一张数据表用于存储用户的信息。数据表结构如表 4-1 所示。

表 4-1 user 信息表

字段名	类型	长度	主键	允许为空	含义
user_id	int	4	是	否	用户 id
username	varchar	18	否	否	用户名
password	varchar	32	否	否	密码
email	varchar	255	否	是	邮箱
phone	char	11	否	是	电话
nickname	varchar	32	否	是	昵称

(2) 后台管理系统需要管理员账号信息进行登录才能进入，管理员信息表是对管理员信息进行存储的。表结构如表 4-2 所示。

表 4-2 admin 信息表

字段名	类型	长度	主键	允许为空	含义
id	int	4	是	否	编号
username	varchar	18	否	否	登录名
password	varchar	32	否	否	密码

(3) 酒店信息多种多样，这时就需要建立存储酒店信息的数据基表。表结构如表 4-3 所示。

表 4-3 hotel 信息表

字段名	类型	长度	主键	允许为空	含义
hotel_id	int	4	是	否	酒店 id
hotel_image	varchar	255	否	否	酒店图片
hotel_name	varchar	32	否	否	酒店名字
hotel_address	varchar	32	否	否	酒店地址
hotel_describe	varchar	255	否	否	酒店描述
hotel_status	int	1	否	否	0 是启用状态，1 是禁用状态
create_date	date	16	否	否	酒店创建时间
single_price	int	8	否	否	单人房价格
double_price	int	8	否	否	双人房价格
big_price	int	8	否	否	大床房价格

(4) 景区旅游系统有各个景点的信息，此时就需要建立一张存储各景点信息的数据表。表结构如表 4-4 所示。

表 4-4 attractions 信息表

字段名	类型	长度	主键	允许为空	含义
attractions_id	int	4	是	否	景点 id
attractions_image	varchar	255	否	否	景点图片
attractions_name	varchar	32	否	否	景点名字
attractions_address	varchar	32	否	否	景点地址
attractions_describe	varchar	255	否	否	景点描述
attractions_status	int	1	否	否	0 是启用状态, 1 是禁用状态
create_date	datetime	32	否	否	创建时间
attractions_price	int	8	否	否	景点价格

(5) 旅游攻略信息也需进行存储。数据信息包括主键攻略 id, 外键用户 id 攻略标题, 攻略描述等。表结构如表 4-5 所示。

表 4-5 strategy 信息表

字段名	类型	长度	主键	允许为空	含义
strategy_id	int	4	是	否	旅游攻略 id
user_id	varchar	10	外键	否	添加该攻略的用户 id
strategy_title	int	255	否	否	攻略标题
strategy_describe	varchar	255	否	否	攻略描述
create_date	date	32	否	否	创建时间
error_message	varchar	255	否	否	审核未通过原因
strategy_status	int	4	否	否	0 是审核通过, 1 是未审核, 2 是审核未通过

(6) 系统还设有景区旅游路线模块，需要设置数据表对旅游路线信息进行存储。如表 4-6 所示。

表 4-6 route 信息表

字段名	类型	长度	主键	允许为空	含义
route_id	int	4	是	否	游玩路线 id
route_name	varchar	32	否	否	路线名称
route_describe	varchar	255	否	否	路线描述
route_status	int	1	否	否	0 是启用状态；1 是禁用状态
route_address	varchar	255	否	否	路线地址
create_date	date	32	否	否	创建时间
update_date	date	32	否	否	修改时间

(7) 用户要对选中的酒店进行预订操作，需要建立一张预订信息数据表存储用户的预定订单。表结构如表 4-7 所示。

表 4-7 user_attractions 信息表

字段名	类型	长度	主键	允许为空	含义
id	int	4	是	否	预订景点 id
user_id	int	4	外键	否	用户 id
attractions_id	int	4	外键	否	景点 id
create_date	datetime	32	否	否	创建时间
ticket_number	varchar	32	否	否	门票号
ticket_num	int	8	否	否	门票数量

phone	varchar	32	否	否	游客电话
order_price	int	32	否	否	订单价格
status	int	1	否	否	订单状态（0 未使用； 1 已使用）

（8）用户登录成功后可以对景点进行预订，user_hotel 信息表用于存放用户对酒店预订的相关信息。数据库字段包括主键 id，外键 user_id, attractions_id, 预订时间，预订电话，价格等。表结构如表 4-8 所示。

表 4-8 user_hotel 信息表

字段名	类型	长度	主键	允许为空	含义
id	int	4	是	否	预订酒店 id
user_id	int	4	外键	否	用户 id
hotel_id	int	4	外键	否	酒店 id
order_price	int	8	否	否	订单价格
arrive_date	date	32	否	否	入住时间
leave_date	date	32	否	否	离店时间
customer_name	varchar	32	否	否	入住人姓名
customer_phone	varchar	32	否	否	入住人电话
customer_card	varchar	32	否	否	入住人身份 证
customer_message	varchar	255	否	否	入住留言
create_date	datetime	32	否	否	预订时间
house_type	varchar	32	否	否	预订的房型

(9) 用户登录后也可以对喜欢的攻略进行收藏关注，表结构如表 4-9 所示。

表 4-9 user_route 信息表

字段名	类型	长度	主键	允许为空	含义
id	int	4	是	否	用户关注的路线 id
user_id	int	4	外键	否	用户 id
route_id	int	4	外键	否	路线 id
create_date	date	32	否	否	关注时间

(10) 用户登录后也可以对景区中感兴趣的旅游路线进行收藏关注。其中字段包括主键 id，外键 user_id，route_id 和 create_date 等等。表结构如表 4-10 所示。

表 4-10 user_strategy 信息表

字段名	类型	长度	主键	允许为空	含义
id	int	4	是	否	用户关注的攻略 id
user_id	int	4	外键	否	用户 id
strategy_id	int	4	外键	否	攻略 id
create_date	date	32	否	否	关注时间

(11) 酒店工作人员需要进行登录后台的酒店管理员系统，需要一张表进行存放酒店人员的账号信息。表结构如表 4-11 所示。

表 4-11 hotel_admin 信息表

字段名	类型	长度	主键	允许为空	含义
id	int	4	是	否	酒店人员 id
hotel_id	int	4	外键	否	酒店 id
username	varchar	18	否	否	用户名
password	varchar	32	否	否	密码

5 系统详细设计与实现

5.1 前台用户系统功能模块设计与实现

5.1.1 用户注册

注册界面展示图如 5-1 所示：



The image shows a registration form titled "注册账号" (Register Account). It contains four input fields: "请输入您的用户名" (Please enter your username) with a user icon, "请输入密码" (Please enter password) with a lock icon, "请再次输入密码" (Please re-enter password) with a lock icon, and "请输入验证码" (Please enter verification code). To the right of the verification code field is a graphic with the characters "A 9 L" and a plus sign. Below the fields is a green "注册" (Register) button and a link "已有账号? 去登录" (Already have an account? Go to login).

图 5-1 注册界面展示图

用户进入系统时可以浏览首页的信息，但需要进行预订或关注等其他操作时都会引导其进行注册登录操作。注册流程图如 5-2 所示：

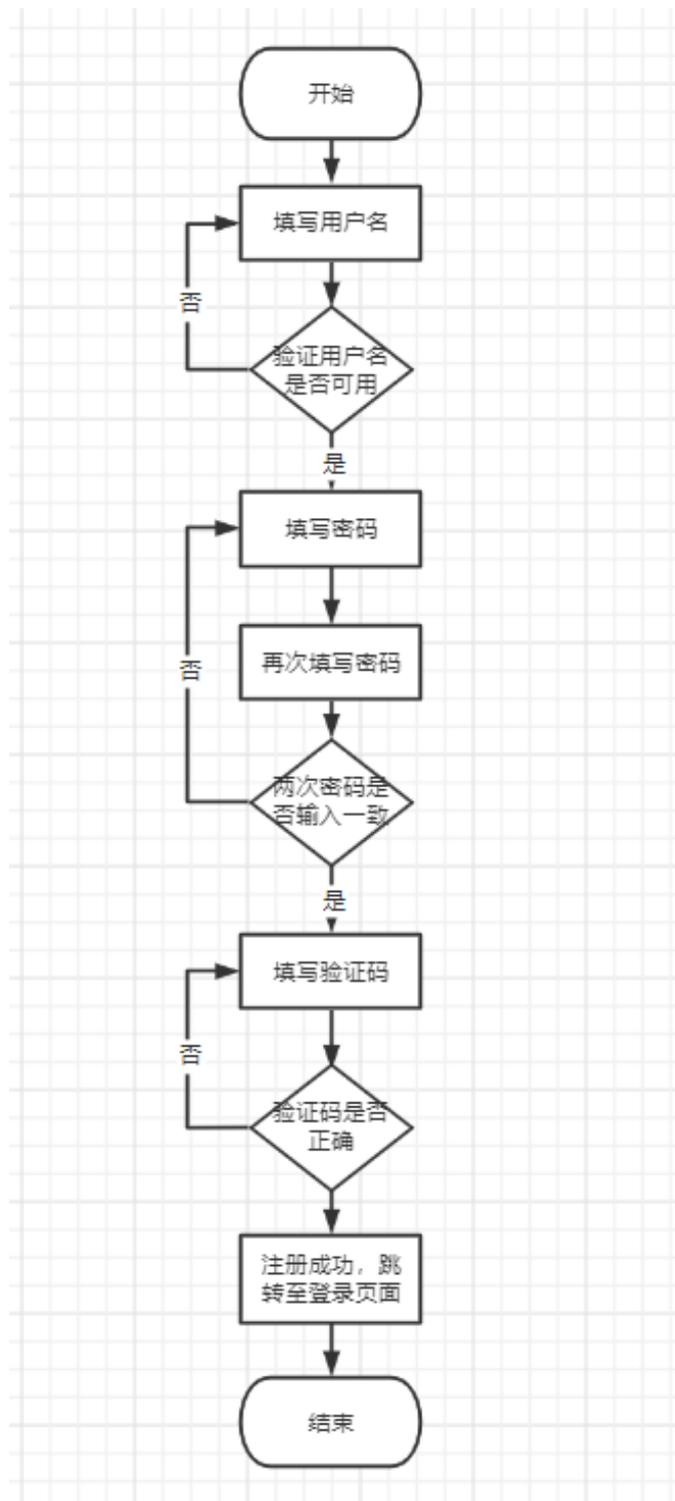


图 5-2 用户注册流程图

5.1.2 用户登录

登录界面展示图如 5-3 所示：



图 5-3 用户登录流程图

用户输入用户名后，将会发 ajax 请求验证该用户名是否存在，不存在则需重新输入，输入密码和验证码后，系统会先判断验证码是否正确，若正确，将会把用户输入的用户名和密码传到后台接口，后台进行查表后，若有查询到数据则判断用户登录成功，并且把用户信息存入 Cookie 中，跳转至系统首页。用户登录流程图如 5-4 所示：

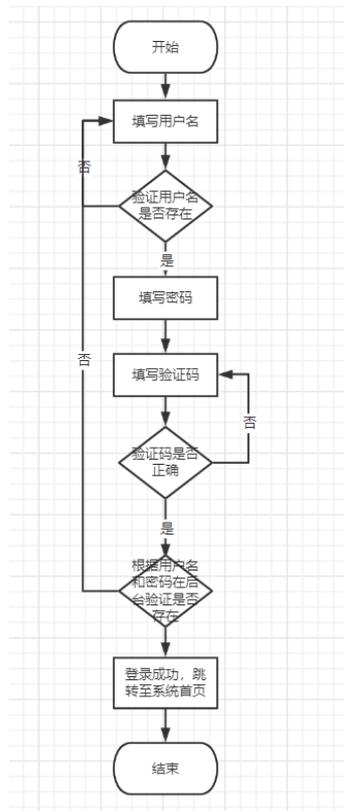


图 5-4 用户登录流程图

5.1.3 个人信息编辑

用户中心界面展示如图 5-5（a），5-5（b）所示：

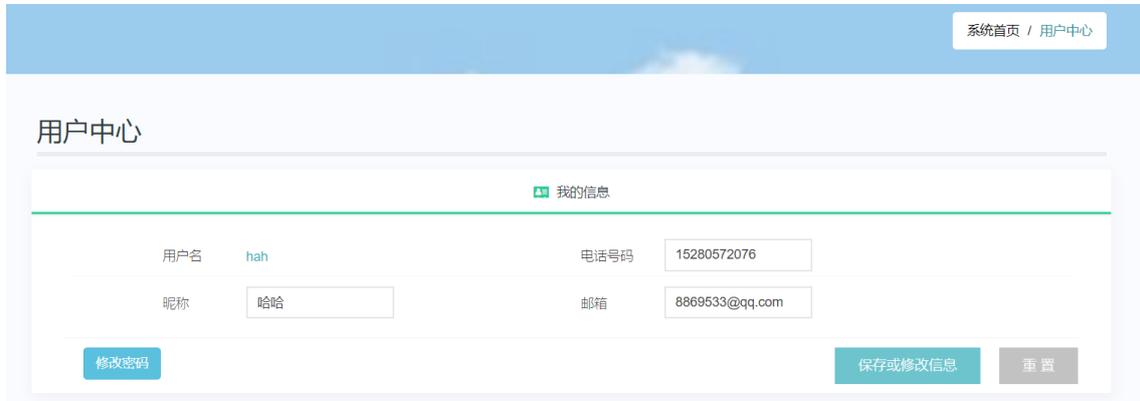


图 5-5（a）用户中心页面展示图



图 5-5（b）用户中心-修改密码页面展示图

用户在登录系统后，在用户中心，其中修改密码在用户输入原密码后则会判断原密码是否正确，正确后才能设置新密码。后台获取新密码后将会进行 MD5 加密，将加密后的密码存入数据库中，代码逻辑如图 5-6 所示。用户个人中心需要对电话、昵称和邮箱等信息进行完善或修改，提交后将会保存用户信息。

```

/**
 * 用户修改密码
 */
@PostMapping("/savePwd")
@ResponseBody
public Msg savePwd(HttpServletRequest request,String oldPwd,String newPwd){
    // 将密码进行md5加密
    String oldPwdMd5 = DigestUtils.md5DigestAsHex(oldPwd.getBytes());
    String newPwdMd5 = DigestUtils.md5DigestAsHex(newPwd.getBytes());
    Cookie cookie = CookieUtil.get(request, name: "username");
    User user = userService.queryUserByUsername(cookie.getValue());
    Integer nums = userService.updatePwd(user.getUserId(), oldPwdMd5, newPwdMd5);
    if (nums == 1){
        return Msg.success();
    }else {
        return Msg.fail();
    }
}
}

```

5-6 用户修改密码代码片段图

5.1.4 预订酒店

在用户系统首页中，会展示部分热门酒店，如图 5-7（a）所示。点击更多酒店，会遍历展示系统所有启用的酒店，在该页面可以进行酒店筛选，查看酒店详情，如图 5-7（b）所示。选中酒店后，会将该酒店 id 传入后台对 hotel 表进行查询后跳转至酒店的详细信息，在此页面可对酒店进行预订操作，如图 5-7（c）所示。



图 5-7（a） 系统首页热门酒店展示图

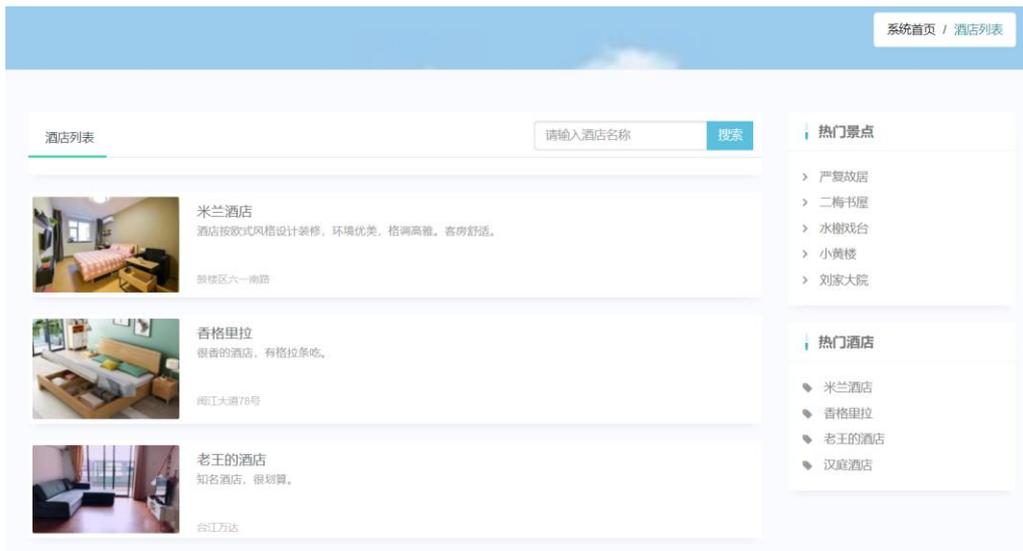


图 5-7 (b) 酒店列表页面展示图



图 5-7 (c) 酒店详情预订页面展示图

酒店预订是系统的主要功能之一，在酒店详情页面，有对该酒店进行预订的按钮，酒店预订页面如图 5-8 所示：

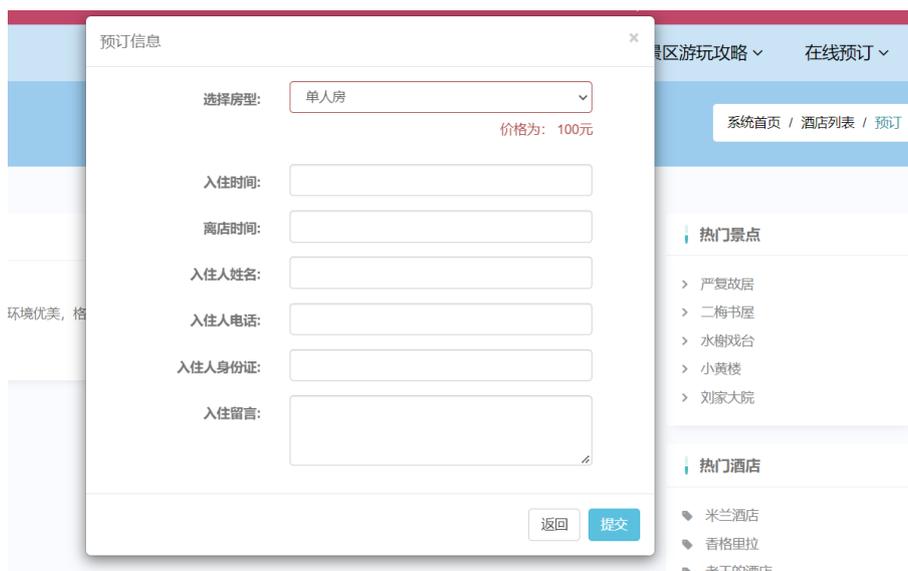


图 5-8 酒店预订表单展示图

用户填入表单相关信息后即会生成预订订单，部分代码逻辑如 5-9 所示：

```
@RequestMapping("/saveReserve")
@ResponseBody
public Msg queryHousePrice(UserHotel userHotel,HttpServletRequest request){
    Cookie cookie = CookieUtil.get(request, name: "username");
    if (cookie == null) {
        throw new ServiceException("用户没有登录!");
    }
    User user = userService.queryUserByUsername(cookie.getValue());
    userHotel.setUserId(user.getUserId());

    if ("单人房".equals(userHotel.getHouseType())) {
        Integer singlePrice = hotelService.querySinglePriceByHotelId(userHotel.getHotelId());
        userHotel.setOrderPrice(singlePrice);
        Integer nums = userHotelService.addUserHotel(userHotel);
        if (nums == 1){
            return new Msg( code: 200, message: "SUCCESS", data: null);
        }else {
            return new Msg( code: 400, message: "预定失败!", data: null);
        }
    }
}
```

图 5-9 用户预订酒店部分代码逻辑图

此订单需要该酒店的工作人员进行审核，在未审核前，用户可随时进行取消预订。预订酒店的流程图如 5-10 所示：

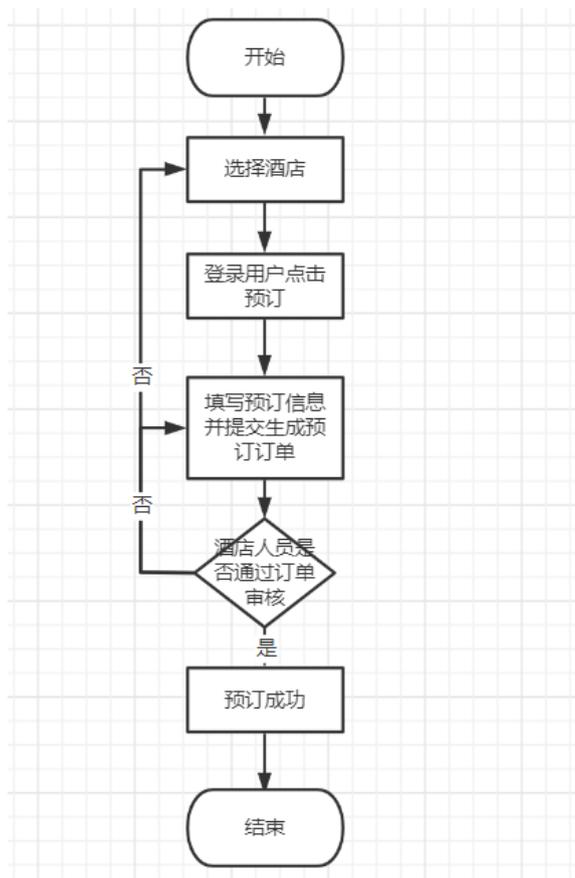


图 5-10 酒店预订流程图

5.1.5 预订景点

景点预订与购买门票大同小异，用户登录系统后，后端程序通过对景点表进行查询存入 request 域中，传给前端做遍历展示所有收费景点信息到首页中，如图 5-11 (a)。用户选中相关景点信息后，后端会把该景点 id 进行查询该景点的详细信息返回给前端，在景点详情中，有预订按钮可对景点进行预订操作，如图 5-11 (b)：

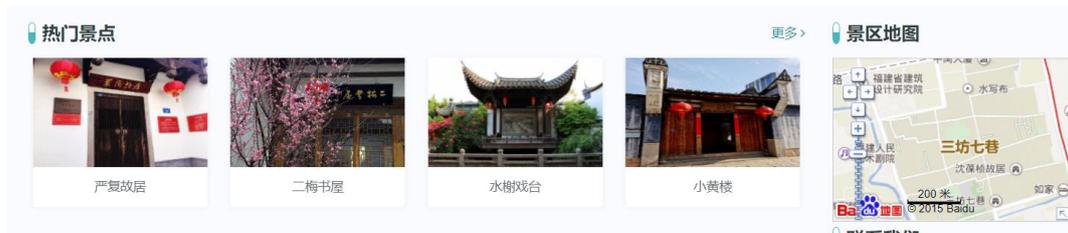


图 5-11 (a) 热门景点展示图



图 5-11 (b) 景点详情展示图

用户点击预订后，输入需要预订的门票，前端发起 ajax 异步请求计算订单价格并局部刷新页面，点击提交后后端会随机生成唯一的门票号，将订单信息存入表中，预订展示图如图 5-11 (c) 所示，代码逻辑如图 5-12 所示，景点预订流程图如 5-13 所示：



图 5-11 (c) 景点预订展示图

```

@RequestMapping("/saveReserve")
@ResponseBody
public Msg saveReserve(HttpServletRequest request, UserAttractions userAttractions) {
    Cookie cookie = CookieUtil.get(request, name: "username");
    if (cookie == null) {
        throw new ServiceException("用户没有登录!");
    }
    User user = userService.queryUserByUsername(cookie.getValue());
    userAttractions.setUserId(user.getUserId());
    userAttractions.setTicketNumber(UUID.randomUUID().toString());
    Integer attractionsPrice = attractionsService.queryAttractionsPriceByAttractionsId(userAttractions.getAttractionsId());
    Integer orderTotalPrice = attractionsPrice * userAttractions.getTicketNum();
    userAttractions.setOrderPrice(orderTotalPrice);
    Integer nums = userAttractionsService.addUserAttractions(userAttractions);
    if (nums == 1){
        return new Msg( code: 200, message: "SUCCESS", data: null);
    }
}

```

图 5-12 景点预订部分代码逻辑图

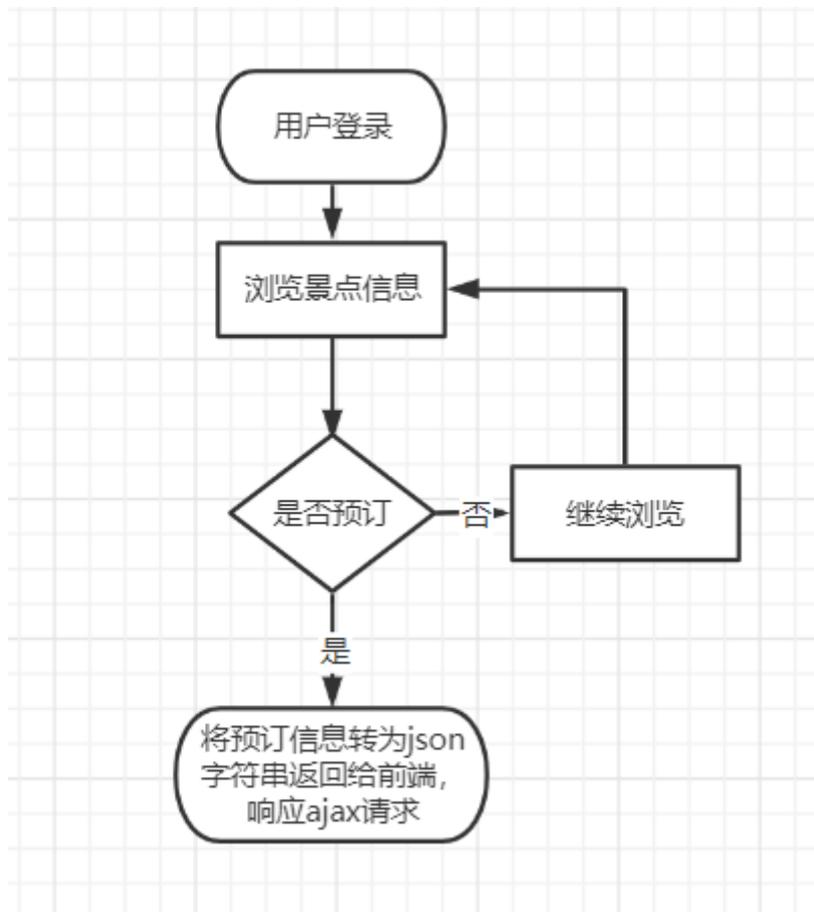


图 5-13 景点预订流程图

5.1.6 我的预订

在系统我的预订中，用户可以查看到自己的酒店和景点预订订单，通过 Cookie 域中取到登录的用户信息，将用户 id 传入酒店预订表和景点预订表中进行查询返

回给前端展示。我的预订页面如图 5-14 所示：



图 5-14 我的预订页面展示图

在订单页面，可以查看自己的预订信息和当前订单状态，酒店订单如是管理员审核状态时，用户可以对该订单进行取消操作，即将该订单 id 进行逻辑删除，取消订单流程图如图 5-15：

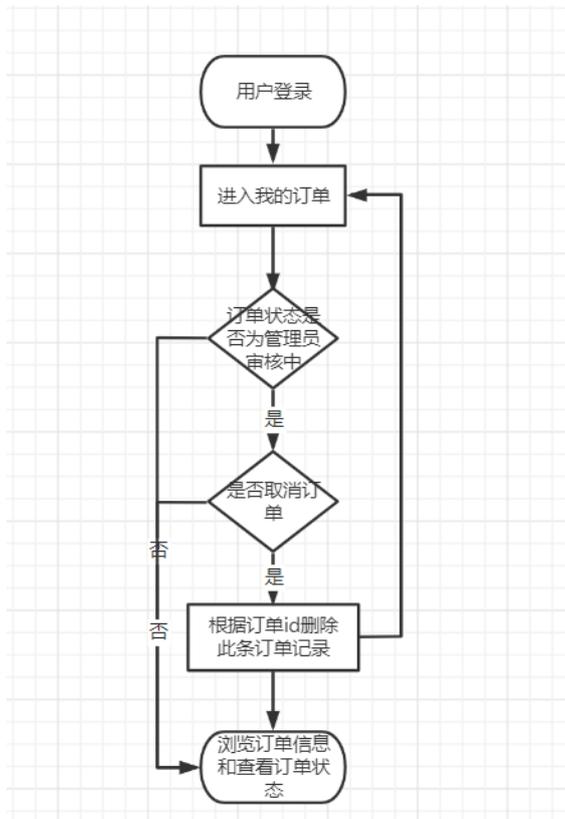


图 5-15 取消订单业务流程图

5.1.7 发布攻略

发布攻略的页面展示如图 5-16 所示：

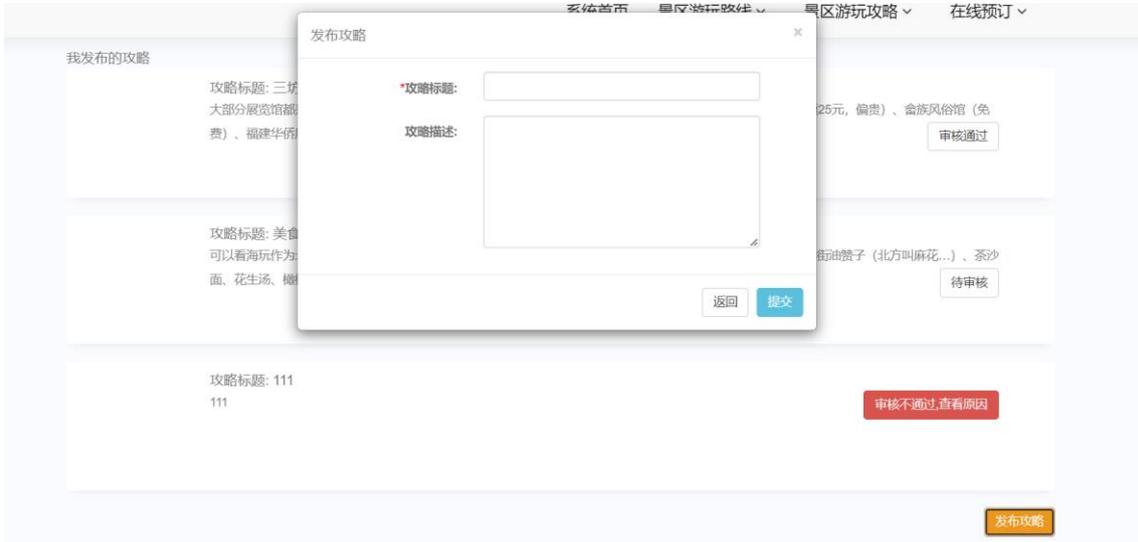


图 5-16 发布攻略页面展示图

用户登录后可以进行攻略的发布，后端获取登录的用户 id，和表单中的攻略标题、攻略描述和默认状态 1 插入数据表中，用户发布的攻略需要管理员进行审核操作，具体流程如图 5-17 所示：

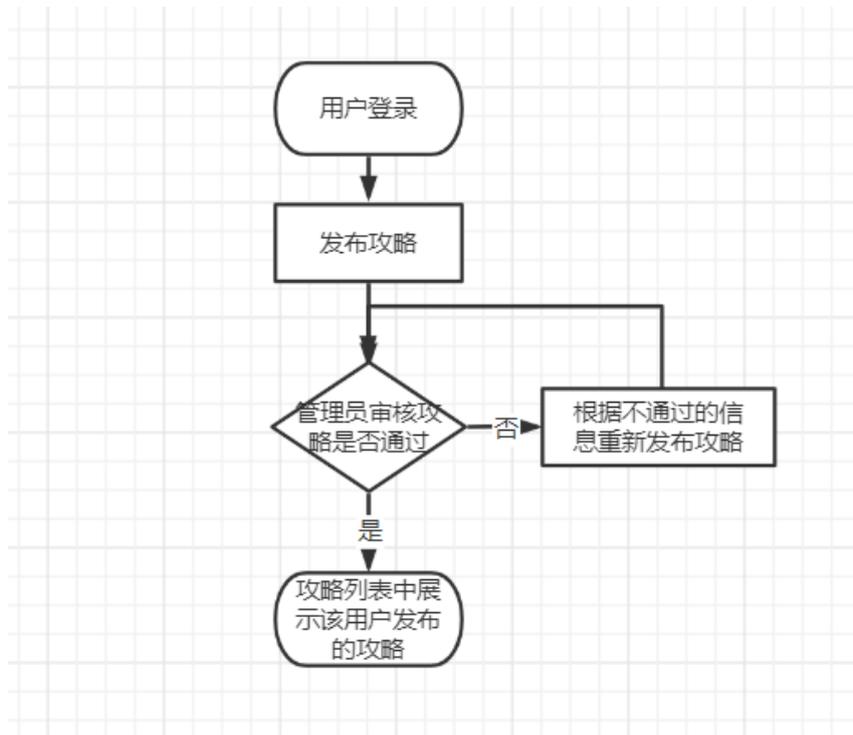


图 5-17 用户发布攻略流程图

5.1.8 关注攻略

景区攻略页面如 5-18 (a) , 5-18 (b) 所示:



图 5-18 (a) 攻略列表展示图



图 5-18 (b) 攻略详情展示图

用户点击攻略详情后判断用户是否关注了此攻略, 若关注过了, 则按钮是取消关注; 若没关注, 按钮是关注。设计逻辑是将用户 id 和攻略 id 传入用户关注的攻略表中进行查询, 看是否有查询结果, 有的话则表示用户已经关注了此攻略, 反之则没关注过, 部分代码逻辑如 5-19 所示:

```

@RequestMapping("/cancelTravelStrategyReserve")
@ResponseBody
public Msg cancelTravelStrategyReserve(HttpServletRequest request, Integer strategyId) {
    Cookie cookie = CookieUtil.get(request, "username");
    if (cookie == null) {
        throw new ServiceException("用户没有登录!");
    }
    User user = userService.queryUserByUsername(cookie.getValue());
    UserStrategy userStrategy = userStrategyService.queryUserStrategyByIdAndStrategyId(user.getUserId(), strategyId);
    if (userStrategy == null) { // 为空是关注 不为空是取消关注
        Integer nums = userStrategyService.addUserStrategy(user.getUserId(), strategyId);
        if (nums == 1) { // 关注成功
            return new Msg( code: 200, message: "SUCCESS", data: null);
        } else {
            return new Msg( code: 400, message: "关注失败!", data: null);
        }
    } else { // 取消关注
        Integer nums = userStrategyService.deleteUserStrategy(user.getUserId(), strategyId);
        if (nums == 1) { // 取消关注成功
            return new Msg( code: 200, message: "SUCCESS", data: null);
        } else {
            return new Msg( code: 400, message: "取消关注失败!", data: null);
        }
    }
}
}

```

图 5-19 用户关注或取消关注部分代码逻辑图

用户关注攻略成功后会跳转至我关注的攻略页面，在该页面前端会遍历展示此用户关注的所有攻略。

5.1.9 关注路线

在景区游玩路线页面中，用户查看路线的详情可以对该路线进行关注或取消关注操作，在点击按钮时，会先通过拦截器判断用户是否登录，若没登录则无法进行操作，会引导用户进行登录，页面展示如图 5-20（a），5-20（b）所示：



图 5-20（a） 路线详情页面展示图



图 5-20（b） 判断用户是否登录页面展示图

关注路线的流程如图 5-21:

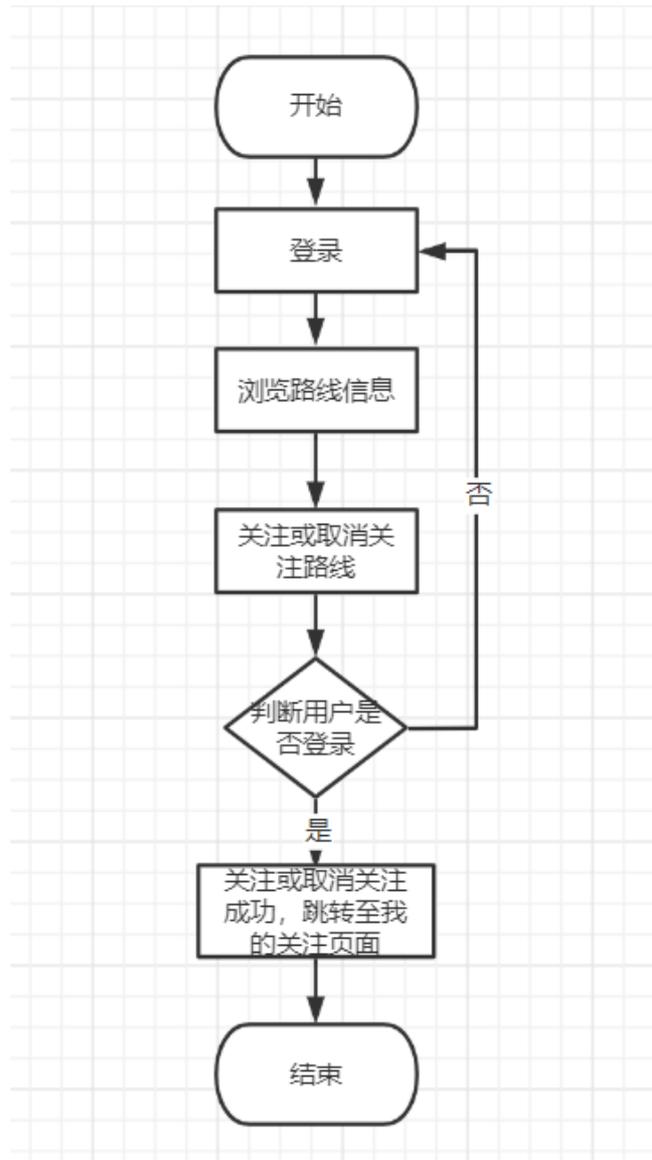


图 5-21 关注路线流程图

5.2 后台管理系统功能模块设计与实现

5.2.1 后台首页模块

后台管理系统以可视化的形式展示系统的用户、酒店等数量，放置了一些常用的按钮可直接跳转至相应功能的界面，给管理员更好的使用体验。其中折线图使用的是 apache 的 echarts 表格，由于其易用性及美观性，Echarts 迅速成为了国内外最常用的可视化前端图表框架之一 [15]，统计用户数、酒店数等是通过 sql 查询表

中的条数进行实时更新显示的。首页数据展示界面如图 5-22 所示：



图 5-22 后台首页数据界面展示图

5.2.2 用户管理模块

(1) 游客用户信息管理

记录了系统的所有游客用户信息，展示用户的用户名、昵称、手机号等信息，通过使用 MyBatis 的 PageHelper 分页插件进行 sql 查询，并以用户 id 升序排列展示，部分代码逻辑如图 5-23 所示。

```
@GetMapping("/userInfo")
public String userInfo(@RequestParam(value = "pageNum", defaultValue = "1") Integer pageNum,
    @RequestParam(value = "pageSize", defaultValue = "5") Integer pageSize,
    Model model) {
    // 引入PageHelper插件，在查询之前调用startPage方法，传入页码以及每页的条数
    PageHelper.startPage(pageNum, pageSize);
    // startPage后的查询就是分页查询
    List<User> userList = userService.queryAllUser();
    // 使用PageInfo包装查询后的结果，并交给页面处理
    // PageInfo封装了详细的分页信息，包括我们查询出来的数据，还可以传入连续显示的页数5[1,2,3,4,5]
    PageInfo<User> pageInfo = new PageInfo<User>(userList, navigatePages: 5);
    model.addAttribute( attributeName: "userInfo",pageInfo);
    model.addAttribute( attributeName: "userList",userList);
}
```

图 5-23 游客用户信息部分代码逻辑图

能够对用于用户信息的修改以及搜索某个用户信息，通过使用 MyBatis 框架执

行对应的 sql 语句实现，对用户信息操作的流程如图 5-24 所示：

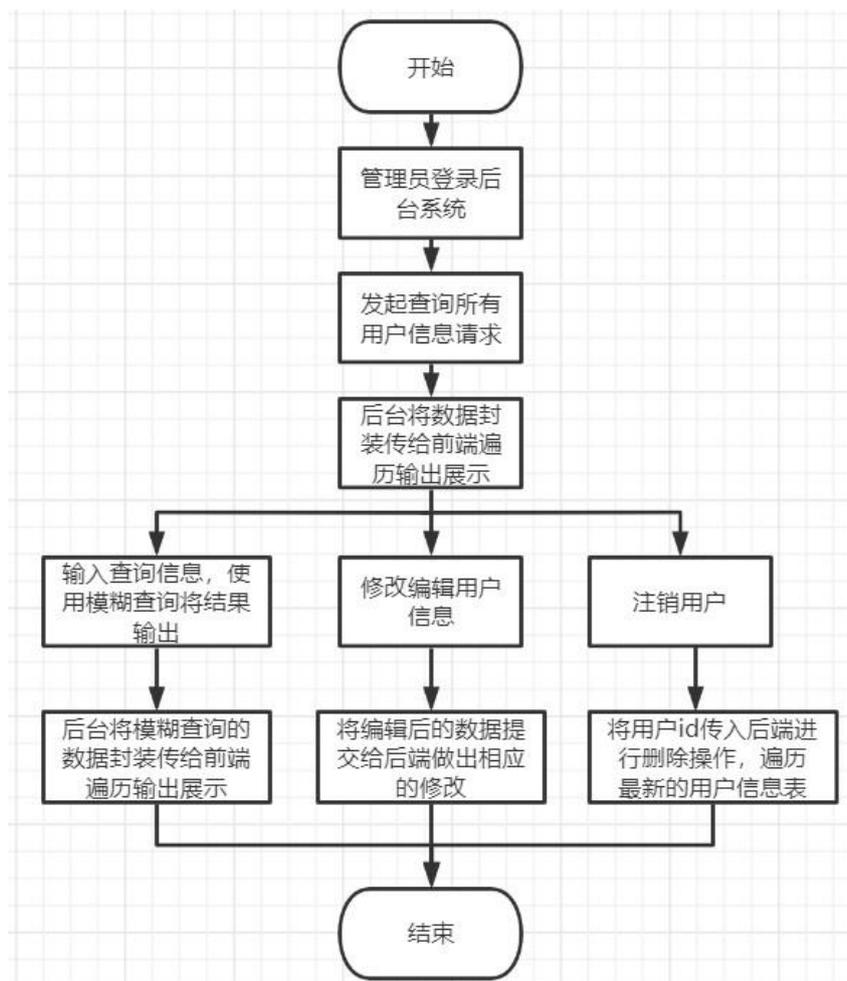


图 5-24 用户信息功能流程图

(2) 酒店人员信息管理

系统为每个酒店配备了一个账号，使用酒店 id 作为外键，用户名、密码作为属性进行酒店账号的存储，在酒店管理端，酒店工作人员需要使用对应的账号进行登录，酒店人员管理的页面展示如图 5-25 所示：

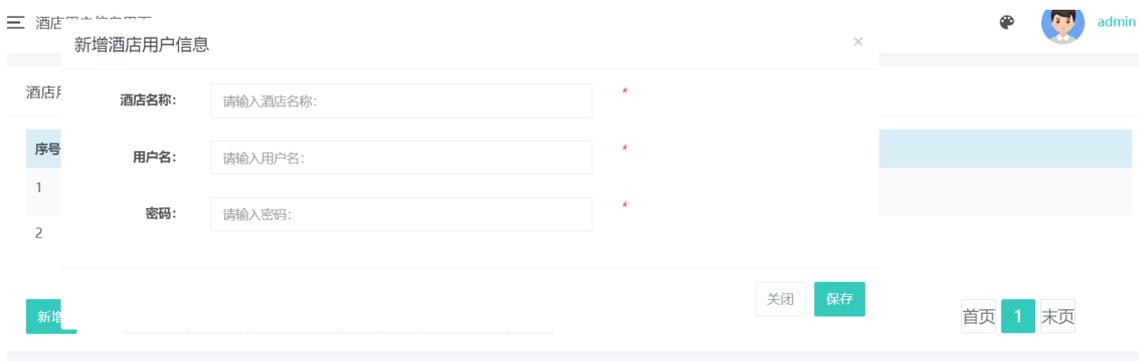


图 5-25 酒店人员页面展示图

输入酒店名称、用户名和密码，将这些作为参数传入后端接口中进行处理，首先根据酒店名称查询出该酒店对应的酒店 id，再将传入的密码进行 MD5 加密保存至数据库中，部分代码逻辑如图 5-26 所示：

```
@PostMapping("/addHotelAdmin")
@ResponseBody
public Msg addHotelAdmin(HotelAdminHotel hotelAdminHotel){
    String pwdMD5 = DigestUtils.md5DigestAsHex(hotelAdminHotel.getPassword().getBytes());
    Integer hotelId = userService.getHotelIdByHotelName(hotelAdminHotel.getHotelName());
    Integer nums = userService.addHotelAdmin(hotelId,hotelAdminHotel.getUsername(),pwdMD5);
    if (nums == 1){
        return Msg.success();
    }
    return Msg.fail();
}
```

图 5-26 新增酒店账户信息部分代码图

5.2.3 酒店管理模块

(1) 所有酒店信息

记录了系统的所有酒店信息，可以根据酒店名、地址等信息进行酒店查询。新增和修改酒店信息是通过前端 form 表单传入用户所输入的信息进行对应的新增和修改操作。其中新增图片的操作流程是先生成一个唯一标识码，再在指定目录创建一个以唯一标识码命名的图片存入本机磁盘中，前端在展示图片时会发起图片请求从而映射到相应的本地磁盘目录中并展示，具体代码如 5-27 (a) (b) 所示：

```
@Configuration
public class ImgConfig implements WebMvcConfigurer {

    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        // 映射图片保存地址
        /*
        http://localhost:8888/images/9fd232c6-2dd0-4a34-9f90-fd90134a8be9.jpg
        映射地址是 D:\file\9fd232c6-2dd0-4a34-9f90-fd90134a8be9.jpg
        */
        registry.addResourceHandler(...pathPatterns: "/images/**").addResourceLocations("file:D:/images/");
    }
}
```

图 5-27 (a) 映射图片的代码图

```

@RequestMapping("/singleUpload")
@ResponseBody
public Map<String, Object> singleUpload(@RequestParam("file") MultipartFile file) {
    Map<String, Object> map = new HashMap<>();
    try {
        String suffixName = file.getOriginalFilename().substring(file.getOriginalFilename().lastIndexOf( str: "."));
        // 生成唯一识别码
        String filename = UUID.randomUUID() + suffixName;
        File filePath = new File(dirPath);
        if (!filePath.exists()) {
            filePath.mkdirs();
        }
        // 创建虚拟路径存储
        simplePath = "http://localhost:8081/images/" + filename;
        map.put("image", simplePath);
        file.transferTo(new File( pathname: dirPath + filename));
        map.put("code", 0);
        map.put("msg", "上传成功");
    } catch (Exception e) {
        map.put("code", 1);
        map.put("msg", "上传失败");
        e.printStackTrace();
    }
    return map;
}

```

图 5-27 (b) 新增图片的代码图

页面中还有一个禁用酒店操作，是将酒店表的 `status` 字段从 0 变成 1 的操作，1 是禁用状态，禁用的酒店在用户端是不会显示的，用户也无法进行对该酒店的预订。酒店信息管理的页面展示如图 5-28 所示：

所有酒店信息

序号	酒店图片	酒店名称	酒店地址	酒店描述	酒店状态	相关操作
1		米兰酒店	鼓楼区六一南路	酒店按欧式风格设计装修，环境优美，格调高雅。客房舒适。	启用	禁用 编辑 删除
2		香格里拉	闽江大道78号	很香的酒店，有格拉条吃。	启用	禁用 编辑 删除
3		阿弥陀佛大酒店	福建省福州奥体	三星酒店，只吃素食。	禁用	启用 编辑 删除

[新增](#)
 页次: 1/2页 每页显示: 3 条 总记录数: 5条

[首页](#)
[1](#)
[2](#)
[»](#)
[末页](#)

图 5-28 酒店信息管理页面展示图

(2) 酒店房型分析

使用 Echarts 表格进行图形可视化，更方便管理员进行查看。通各个酒店用户订单数和所有酒店的房型订单数进行统计，后端通过对酒店订单表的分组查询进行数据统计返回给前端，前端使用 Echarts 框架对数据进行处理展示，页面如图 5-29

所示:

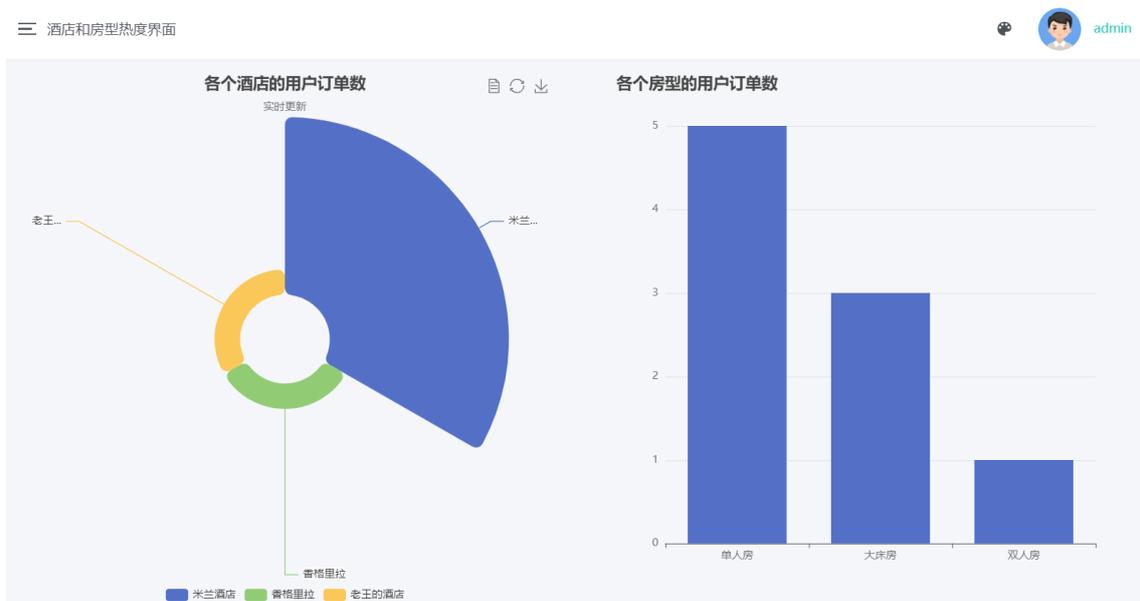


图 5-29 酒店房型分析页面展示图

5.2.4 景点管理模块

(1) 景点信息设置

管理员用户可以对收费景点进行增减、禁用和编辑等操作,添加景点信息时,用户提交信息会发送 Ajax 请求调用后端对数据库进行插入操作。景点信息列表的展示以分页的形式展现,默认显示第一页和每页显示三条记录,使用 PageHelper 插件,后端只需对数据表进行全部查询,将结果封装进 PageInfo 中即可,代码逻辑如 5-30 所示,景点信息页面如 5-31 所示:

```
@RequestMapping("/attractionsInfo")
public String attractionsInfo(@RequestParam(value = "pageNum", defaultValue = "1") Integer pageNum,
    @RequestParam(value = "pageSize", defaultValue = "3") Integer pageSize,
    Model model) {
    // 引入PageHelper插件,在查询之前调用startPage方法,传入页码以及每页的条数
    PageHelper.startPage(pageNum, pageSize);
    // startPage后的查询就是分页查询
    List<Attractions> attractionsList = attractionsService.queryAllAttractions();
    // 使用PageInfo包装查询后的结果,并交给页面处理
    // PageInfo封装了详细的分页信息,包括我们查询出来的数据,还可以传入连续显示的页数5[1,2,3,4,5]
    PageInfo<Attractions> pageInfo = new PageInfo<>(attractionsList, navigatePages: 3);
    model.addAttribute("attractionsPageInfo", pageInfo);
    model.addAttribute("attractionsList", attractionsList);
}
```

图 5-30 景点信息部分代码逻辑图

所有景点信息

序号	景点图片	景点名称	景点地址	景点描述	景点状态	相关操作
1		严复故居	福州市鼓楼区郎官巷西段北侧20号	严复故居，位于福州市鼓楼区郎官巷西段北侧20号，坐北朝南，主座与花厅两座毗连，占地面积609平方米。主座清式规制，门内三面走廊，前设有插屏门。大厅面阔三间，正间分前后厅，左右边间为前后厢房。主座前廊西有小门通花厅。花厅前后走廊、栏杆均仿西方建筑纹饰。	启用	禁用 编辑 删除
2		二梅书屋	福州市鼓楼区郎官巷24号	始建于明末，清光绪及民国间大修。宅占地面积2126平方米，坐南向北，木构，坐南向北，前后、东西共五进。2006年公布为全国重点文物保护单位。	启用	禁用 编辑 删除
3		水榭戏台	福州市三坊七巷景区衣锦坊东口北侧	水榭戏台现作为福州地方戏剧演艺场，经过多次重修，成为三座毗连、全坊最大的宅院。从西而东，第一座为主座大院，第二座为别院，第三座为花园林。花厅的最大特色是建有水榭戏台，是福州市仅存的水榭戏台，它是府内喜庆宴会的重要场所。	启用	禁用 编辑 删除

新增 页次: 1/3页 每页显示: 3 条 总记录数: 7条 首页 1 2 3 » 末页

图 5-31 景点信息页面展示图

(2) 订单状态审核

针对用户对景点的预订，后台管理员需要对这些预订订单进行审核操作，状态有使用和未使用。管理员根据订单信息进行订单审核，若该用户使用了此门票，管理员需将状态改为已使用，即根据订单 id 将数据表的 status 字段改为 1 即可。页面展示如图 5-32 所示：

订单状态审核

序号	预订景点	预订用户	订单价格	预订时间	订单状态	相关操作
1	小黄楼	hah	20	2022-03-23 16:05:22	未使用	查看预订信息 已使用
2	二梅书屋	hah	240	2022-01-30 16:21:18	未使用	查看预订信息 已使用
3	二梅书屋	lisi	360	2022-01-29 21:26:30	未使用	查看预订信息 已使用
4	二梅书屋	xxi	240	2022-01-29 21:25:13	已使用	查看预订信息
5	严复故居	xxi	100	2022-01-29 21:25:04	已使用	查看预订信息

页次: 1/2页 每页显示: 5 条 总记录数: 7条 首页 1 2 » 末页

图 5-32 景点订单状态审核页面展示图

(3) 景点热度分析

根据用户对景点的预订情况，统计各个景点的订单数，对 user_attractions 表进

行分组查询，MyBatis 的 sql 代码如图 5-33 所示：

```
<!-- 查询各个景点的订单 -->
<select id="selectAttractionsHot" resultMap="AttractionsHot">
    SELECT a.attractions_name,count(*) count
    FROM user_attractions ua
         INNER JOIN attractions a
    ON ua.attractions_id = a.attractions_id
    GROUP BY ua.attractions_id
</select>
```

图 5-33 景点热度分析的部分代码图

5.2.5 攻略管理模块

(1) 攻略信息设置

在此页面，其中攻略描述的前端使用的是富文本编辑器，引入了 wangEditor.js 实现富文本编辑框的效果，页面展示如图 5-34，前端代码如 5-35 所示：

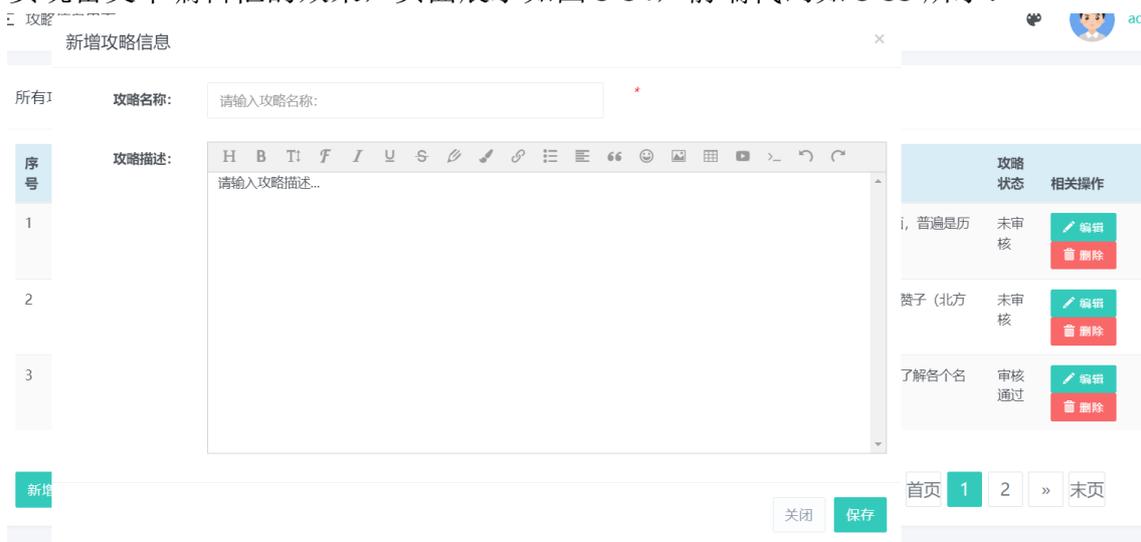


图 5-34 新增修改时的富文本编辑器展示图

```

<script th:src="@{/js/wangEditor.js}"></script>
<!-- 自定义js脚本-->
<script th:src="@{/year/js/main.min.js}"></script>
<script type="text/javascript">

    //点击新增按钮弹出模态框。
    $("#add_modal_btn").click(function () {

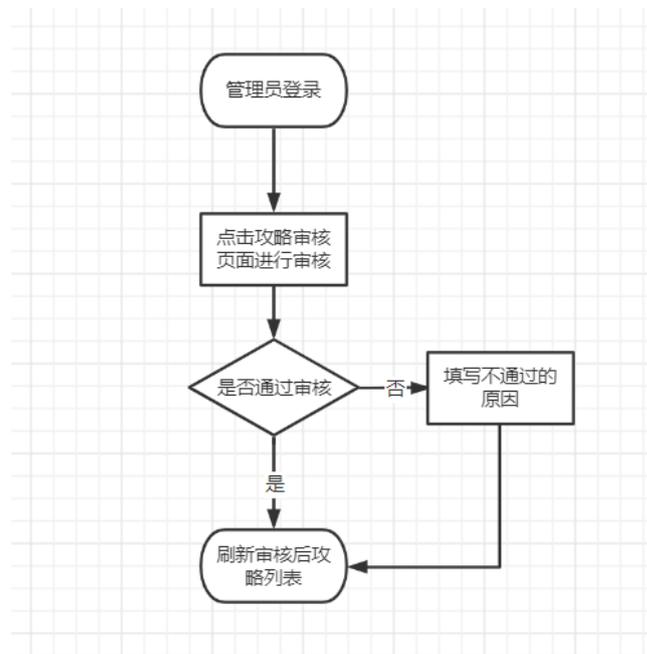
        //限制富文本编辑器在模态框中的显示
        var E = window.wangEditor;
        var editor = new E('#addDescEditor');
        editor.customConfig.zIndex = 100;
        editor.customConfig.onChange = function(html) {
            // 监控变化, 同步更新到 textarea
            //alert(html);
            textArea = html;
            $("#addDescEditor").val(html);
            $("#addStrategyDesc").val(html);
            //alert($("#divEditor").val());
        }
        editor.create();
    });

```

图 5-35 部分前端代码图

(2) 攻略审核

攻略的发布分为管理员发布和用户发布, 管理员发布的攻略不需要进行审核操作。通过审核的攻略则会被展示在前台用户系统中供游客进行关注。管理员攻略审核的流程图如 5-36 所示, 页面展示如图 5-37 所示:



5-36 管理员审核攻略流程图



5-37 攻略审核页面展示图

(3) 用户关注的攻略

管理员可以查看到所有用户所关注的攻略信息，包括攻略名称、关注该攻略的用户名和关注时间等，页面如图 5-38 所示：



5-38 用户关注的攻略页面展示图

5.2.6 路线管理模块

(1) 路线信息设置

管理员可以对所有的路线信息进行设置包括查询，查询模块使用的是模糊查询，用户可以输入部分的路线名称、地址和路线描述等信息进行查询，后端接口将管理员输入的字符串获取到使用 sql 的 Like 关键字进行模糊查询，代码如图 5-39 所

示:

```
<select id="selectRouteByLike" resultMap="BasicMap">
  SELECT route_id,route_name,route_describe,route_status,route_address,create_date,update_date
  FROM route WHERE CONCAT(
    route_name,
    route_address,
    route_describe
  ) LIKE CONCAT("%", #{keywords}, "%") ORDER BY route_id ASC
</select>
```

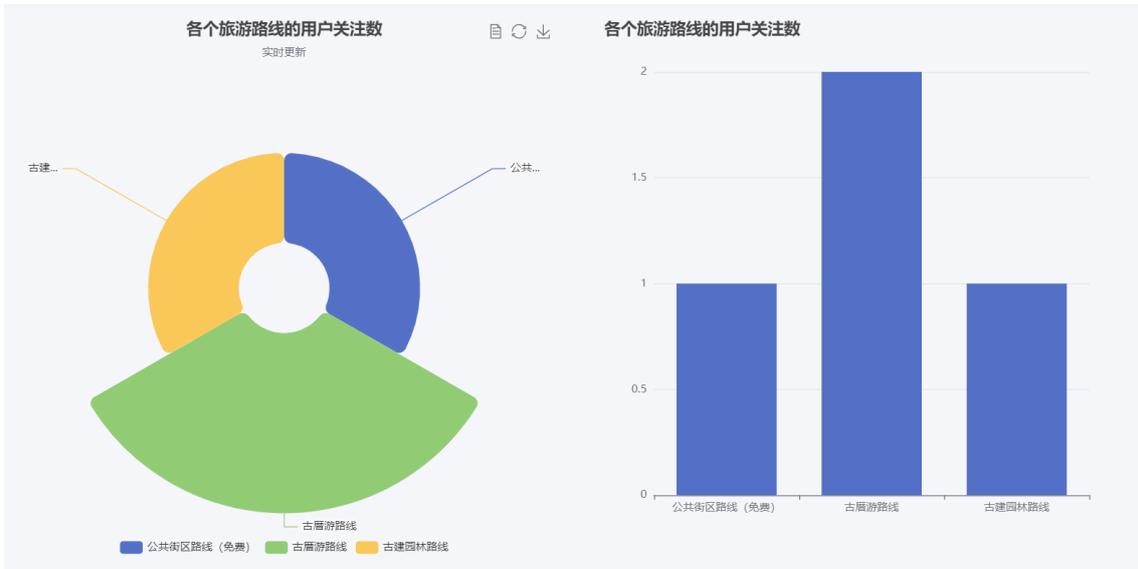
5-39 模糊查询的代码图

(2) 路线热度分析

后端接口通过统计所有路线信息的用户关注数，结合前端的 Echarts 图表工具进行可视化展示，部分前端代码如 5-40 所示，页面图如 5-41 所示：

```
// 柱状图
barOption = {
  title: {
    text: '各个旅游路线的用户关注数'
  },
  tooltip: {
    trigger: 'axis',
    axisPointer: {
      type: 'shadow'
    }
  },
  legend: {
    top: 'bottom'
  },
  xAxis: {
    type: 'category',
    data: []
  },
  yAxis: {
    type: 'value'
  },
  series: [
    {
      data: [],
      type: 'bar'
    }
  ]
};
```

5-40 柱状图的前端部分代码图

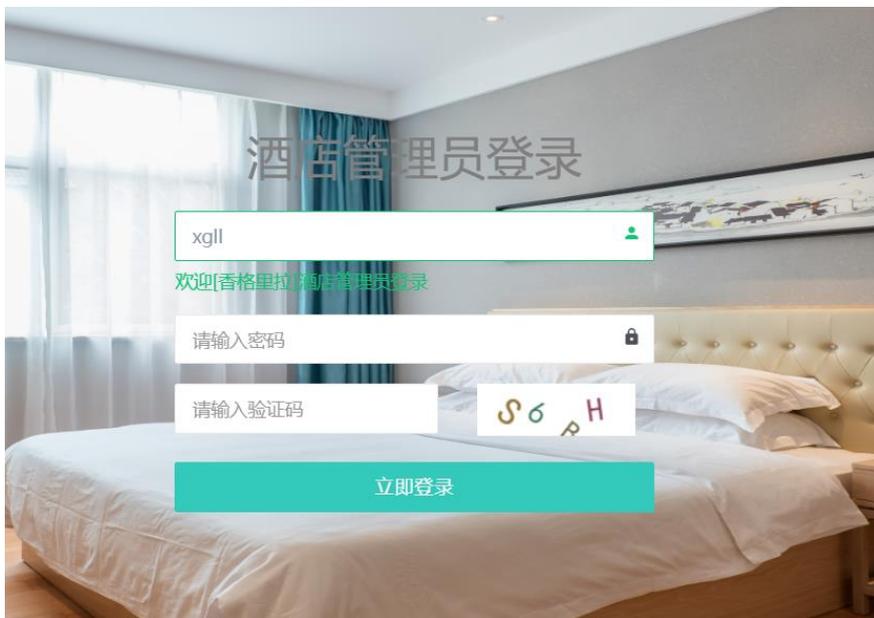


5-41 路线热度分析页面展示图

5.3 后台酒店管理员系统功能模块设计与实现

5.3.1 酒店管理员系统登录

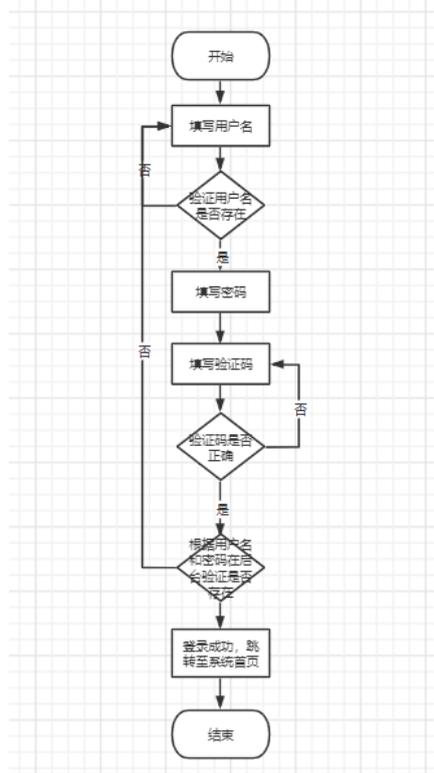
页面展示如图 5-42 所示：



5-42 酒店管理员登录页面展示图

酒店工作人员输入用户名后，前端会发送 Ajax 请求将用户名传入后端中，后端通过查表，若该用户名存在则会返回酒店名称给前端进行局部刷新展示，不存

在则会提示用户名不存在请重新输入！管理员登录的过程流程如图 5-43 所示：



5-43 酒店管理员登录流程图

5.3.2 酒店订单审核

酒店工作人员使用管理员分配给酒店的账号登录酒店管理员系统，后端通过查询得到登录用户的酒店 id，查询 user_hotel 表中对应的 hotel_id 就是属于该酒店的预订订单，该酒店工作人员只能对属于这个酒店的订单进行审核操作，具体的订单状态如表 5-44 所示：

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> 1	预订酒店id
user_id	int	0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	用户id
hotel_id	int	0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	酒店id
create_date	datetime	0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	预订时间
house_type	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	预订的房型
order_price	int	0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	订单价格
arrive_date	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	入住时间
leave_date	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	离店时间
customer_name	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	入住人姓名
customer_phone	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	入住人电话
customer_card	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	入住人身份证
customer_message	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	入住留言
status	int	0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	预订状态: 0预定中; 1预订成功; 2已入住; 3已退房;

5-44 酒店预订订单的状态图

酒店人员通过预订信息进行一步一步审核操作，即后端修改表中 status 字段的

值，前端根据 status 的值进行判断显示输出，每审核一次即从数据库中获取最新的数据信息，页面如图 5-45 所示：

订单状态审核

序号	预订酒店	预订用户	订单价格	预订房型	预订时间	订单状态	相关操作
1	米兰酒店	hah	100	单人房	2022-04-22 11:06:39	预订中,待审核	入住人信息 通过预订 不通过预订
2	米兰酒店	hah	100	单人房	2022-04-19 19:46:56	预订成功,待入住	入住人信息 入住
3	米兰酒店	hah	150	双人房	2022-01-30 16:55:14	已入住,待离店	入住人信息 退房离店
4	米兰酒店	lisi	100	单人房	2022-01-29 21:26:07	预订失败	入住人信息
5	米兰酒店	xxi	333	大床房	2022-01-27 16:33:51	已入住,待离店	入住人信息 退房离店

页次: 1/2页 每页显示: 5 条 总记录数: 7条

首页 1 2 » 末页

5-45 酒店订单审核页面展示图

5.3.3 酒店价格设置

酒店工作人员还可对自家酒店进行房型的价格设置，可以根据季节和旅游热度等情况进行修改酒店的价格，页面展示如图 5-46 所示：

设置修改酒店价格

酒店名称: 米兰酒店

酒店地址: 鼓楼区六一南路

序号	1
单人房价格:	<input type="text" value="100"/>
双人房价格:	<input type="text" value="150"/>
大床房价格:	<input type="text" value="666"/>

相关操作

[修改价格](#)

关闭 修改

5-46 酒店价格设置页面展示图

6 系统测试

6.1 测试目的

软件测试是通过各种测试方法找出软件在功能、设计和业务逻辑上存在的问题，保证用户在使用系统过程中不出问题。软件测试是验证系统有效性和可靠性的重要保障^[16]。

6.2 系统的功能测试

黑盒测试是一种软件测试方法，软件测试员只需知道软件要做什么而无法看到盒子里面的软件是如何运行的，测试人员不需要不在乎产品内部的结构，只需针对产品的功能做出检测评价^{[17][18]}。白盒测试需要全面了解程序的内部逻辑结构，往往需要较高的测试代价^[19]。

6.2.1 前台用户系统功能测试

(1) 系统登录模块测试结果如表 6-1 所示：

表 6-1 系统登录模块测试结果

序号	输入条件	预期输出	实际结果
1	不填任何信息，直接点击登录按钮	提示输入用户名	相同
2	输入不存在的用户名	提示用户名存在，请重新输入	相同
3	输入错误的密码	提示密码错误，请重新输入	相同
4	用户名、密码和验证码都输入正确	登录系统成功	相同

(2) 酒店预订模块测试结果如表 6-2 所示：

表 6-2 酒店预订模块测试结果

序号	输入条件	预期输出	实际结果
1	用户未登录，选择酒店进行预订	提示用户需先进行登录	相同
2	用户已登录，预订表单未全部填写	提示用户某信息不能为空	相同
3	用户已登录，预订表单全部正确填写	预订成功，跳转至我的预订页面	相同

(3) 景点预订模块测试结果如表 6-3 所示：

表 6-3 景点预订模块测试结果

序号	输入条件	预期输出	实际结果
1	用户未登录，选择景点进行预订	提示用户需先进行登录	相同
2	用户已登录，门票数量未填写	提示用户门票数量不能为空	相同
3	用户已登录，门票数量已填写，电话已填写	提示用户联系电话不能为空	相同
4	用户已登录，门票数量和电话都已填写	预订成功，跳转至我的预订页面	相同

(4) 攻略和路线关注模块测试结果图表 6-4 所示：

表 6-4 攻略和路线关注模块测试结果

序号	输入条件	预期输出	实际结果
1	用户未登录，进行关注操作	提示用户需先进行登录	相同
2	用户已登录，对已关注过的攻略或路线进行关注	没有关注按钮，只有取消关注按钮	相同
3	用户已登录，对未关注的攻略或路线进行关注操作	关注成功，跳转至我的关注页面	相同

6.2.2 后台管理员系统功能测试

(1) 用户信息管理模块测试结果图表 6-5 所示:

表 6-5 用户信息管理模块测试结果

序号	输入条件	预期输出	实际结果
1	查询某个用户, 如输入用户名 hah	查询到用户名为 hah 的用户信息	相同
2	注销用户	用户信息表中有该用户的信息	相同
3	编辑用户信息, 如将电话修改为 15280572222	手机号变为 15280572222	相同

(2) 酒店信息管理模块测试结果如表 6-6 所示:

表 6-6 酒店信息管理模块测试结果

序号	输入条件	预期输出	实际结果
1	查询某个酒店, 如输入酒店名米兰	查询到酒店名为米兰酒店的酒店信息	相同
2	新增酒店信息	酒店信息表中有多出了新增的酒店信息	相同
3	禁用酒店	被禁用的酒店状态为禁用中	相同
4	删除酒店信息	酒店信息从信息表中消失	相同
5	编辑酒店信息, 如将米兰酒店地址改为七一北路	米兰酒店地址变为七一北路	相同
6	启用酒店	被启用的酒店状态为启用	相同

(3) 景点信息管理模块测试结果如表 6-7 所示:

表 6-7 景点信息管理模块测试结果

序号	输入条件	预期输出	实际结果
1	查询某个景点, 如输入景点名严复	查询到景点名为严复故居的景点信息	相同
2	新增景点信息	景点信息表中有多出了新增的景点信息	相同
3	禁用景点	被禁用的景点状态为禁用中	相同
4	删除景点信息	景点信息从信息表中消失	相同
5	编辑景点信息, 如将严复故居描述改为好玩	描述变为好玩	相同
6	启用景点	被启用的景点状态为启用	相同
7	将用户订单状态设置为已使用	订单状态展示为已使用	相同

(4) 攻略信息管理模块测试结果如表 6-8 所示:

表 6-8 攻略信息管理模块测试结果

序号	输入条件	预期输出	实际结果
1	查询某个攻略, 如输入: 美食	查询到南后街和美食两条旅游攻略	相同
2	新增攻略信息	景点信息表中有多出了新增的攻略信息	相同
3	通过攻略信息审核	攻略审核状态为审核通过	相同
4	删除攻略信息	攻略信息从信息表中消失	相同
6	重新审核攻略新	攻略审核状态为未审核	相同

(5) 路线信息管理模块测试结果如表 6-9 所示：

表 6-9 路线信息管理模块测试结果

序号	输入条件	预期输出	实际结果
1	查询某个路线，如输入：免费	查询到一条免费的路线新	相同
2	新增路线信息	路线信息表中有多出了新增的路线信息	相同
3	禁用路线信息	被禁用的景点状态为禁用中	相同
4	删除路线信息	路线信息从信息表中消失	相同
5	编辑路线信息，如将免费路线修改为买	描述变为买	相同
6	启用路线	被启用的路线状态为启用	相同

6.2.3 后台酒店人员系统功能测试

(1) 酒店订单审核模块测试结果如表 6-10 所示：

表 6-10 酒店订单审核模块测试结果

序号	输入条件	预期输出	实际结果
1	用户点击不通过预订	订单状态为预订失败	相同
2	用户点击通过预订	订单状态为预订成功,待入住	相同
3	用户点击入住	订单状态为已入住,待离店	相同
4	用户点击退房离店	订单状态为已离店	相同

(2) 酒店价格设置模块测试结果如表 6-11 所示:

表 6-11 酒店价格设置模块测试结果

序号	输入条件	预期输出	实际结果
1	分别输入单人房、双人房和大床房价格	价格修改为刚刚输入的价格	相同
2	点击酒店介设置	显示该酒店各个房型的价格	相同

6.3 测试总结

通过软件测试,使用不同的测试用例,验证了软件的安全、稳定及可靠性,确保系统是准确无误的^[20]。

7 总结与展望

本篇论文详细介绍了三坊七巷景区旅游系统的设计与实现，该系统基于 SpringBoot 框架，使用 MySQL 和 Thymeleaf 等技术共同开发。通过这次毕业设计，完整的开发了一个系统，从需求分析、数据库设计到系统设计，整个开发流程学习了许多新的技术和框架。系统运行时，用户和管理员均可以正常使用，具有一定的可靠性和安全性。

系统经过软件测试可以验证系统的可靠性，系统设计之初就考虑了系统的可扩展性，便于之后有更好的想法可以继续完善和升级。目前市面上暂无针对三坊七巷景区的旅游系统，该系统能够很好的填补这一空白，为游客和景区提供更方便的服务。本人开发经验有限，系统的功能还不是足够完善，还有待日后逐步改进优化。

参考文献

- [1] 高立慧, 李洪波. 基于游客参与的文化创意旅游街区游客忠诚影响研究——基于福州三坊七巷的实证研究[J]. 资源开发与市场, 2019, 35(06):875-883.
- [2] 陈少辉. 福州市三坊七巷历史文化街区的保护与旅游开发研究[D]. 广西师范大学, 2021.
- [3] 王祎霖. 基于 JSON 的 Web 服务描述框架的设计与实现[D]. 天津大学, 2014.
- [4] 陈玥. 基于 Java 语言开发的宿舍管理系统[J]. 信息与电脑(理论版), 2021, 33(21):38-40.
- [5] 李旭. 基于 Java 语言的操作系统设计与实现技术研究[D]. 国防科学技术大学, 2008.
- [6] 陈峰. 基于 SSM 框架的 B2C 网上商城系统的设计与实现[D]. 湖南大学, 2018.
- [7] 刘靖桐. 面向 Web2.0 的 web 应用前端开发框架的设计与实现[D]. 北京邮电大学, 2014.
- [8] IntelliJ IDEA 2016.2 功能升级[J]. 电脑编程技巧与维护, 2016(14):4.
- [9] 余炆, 曲毅, 孙亦乐. 基于 Apache Tomcat 的一站式 Java 应用服务器解决方案[J]. 中国金融电脑, 2018(01):59-63.
- [10] 张欢庆. 基于 ARM 处理器的嵌入式 Linux 系统关键技术研究[D]. 山东师范大学, 2013.
- [11] 石怡. 基于 MySQL 数据库的查询性能优化研究[J]. 四川职业技术学院学报, 2021, 31(01):164-168.
- [12] 周紫钰, 刘越. 基于 B/S 的核酸检测信息管理系统的设计与开发[J]. 山西电子技术, 2021(03):55-58.
- [13] 原方亮. 基于 Bootstrap 的 H5 响应式网站开发技术研究[D]. 郑州大学, 2018.
- [14] 林立疆. 基于移动互联网的景区智慧旅游系统设计与实现[D]. 南京邮电大学, 2019.
- [15] 刘翔宇. 基于 Vue 的数据可视化系统的设计与实现[D]. 北京邮电大学, 2018.
- [16] 杨伟海. 在线考试的智能题库管理系统的设计与实现[D]. 北京邮电大学, 2019.
- [17] 余慧敏, 徐白, 周楷林, 等. 动态软件测试中的白盒测试和黑盒测试探讨[J]. 电子测试, 2018, (08):58-59.
- [18] 妥泽花. 基于黑盒测试与白盒测试的比较探究[J]. 电子世界, 2021, (11):55-56.
- [19] 赵一品. 基于 Spring Boot 和 MyBatis 的银行知识库管理系统的设计与实现[D]. 山东大学, 2020.
- [20] 余慧敏, 徐白, 周楷林, 等. 动态软件测试中的白盒测试和黑盒测试探讨[J]. 电子测试, 2018, (08):58-59.

致谢

毕业设计的完成也意味着大学生涯即将结束，四年时间转瞬即逝，感叹时光匆匆，大学过往皆历历在目，目光所及都是最美好的回忆。

感谢我的指导老师曹永忠教授，毕业设计的完成离不开曹老师的用心指导，曹老师认真负责的态度让我十分敬佩，祝愿曹老师身体健康，万事胜意！感谢学院的所有老师，授予我专业知识和做人的道理。感谢父母和朋友，给我最长久的陪伴。感谢 407 的所有舍友，四年的相处让我们一起进步。感谢所有每一位关心过、帮助过我的人。

最后，愿我们乘风破浪，前程似锦！