

# 湖南工业大学科技学院

## 课程设计资料

课程名称: Python 程序设计课程设计

设计题目: 基于 Django 框架 Shine She 旅游系统

专 业: 计算机科学与技术 班级: 计科 2106

学生姓名: 刘继友 学 号: 21412030655

指导教师: 肖 哲

### 材 料 目 录

序号	名称	数量
1	课程设计任务书	1
2	课程设计报告书	1
3	设计文档及程序源代码(电子档)	1



湖南工业大学科技学院  
COLLEGE OF SCIENCE AND TECHNOLOGY HNUT

# 课程设计任务书

课程名称: Python 程序设计课程设计

设计题目: 基于 Django 框架 Shine She 旅游系统

专 业: 计算机科学与技术 班级: 计科 2106

学生姓名: 刘继友 学 号: 21412030655

起止日期: 2024 年 5 月 22 日 ~2024 年 6 月 2 日

指导教师: 肖 哲

# 湖南工业大学科技学院教务部 制课程设计任务书

## 1. 课程设计的内容和要求（包括原始数据、技术要求、工作要求等）：

### 内容：

分析 Shine She 旅游系统，得到游客和管理员的操作流程，游客首先创建自己的普通用户账号，进入系统可以看到首页界面，里面有许多景区可以选择，并且可以看到对应的评论个数、景区的等级、前十的评分景区，然后可以在左侧导航栏中看到个人中心，里面包括修改信息、修改密码。之后是购票中心，里面包括购票和购票详情页面，可以在购票界面看到对应的添加信息，添加完成点击确认，去往购票详情页，可以看到购票信息并且修改购票信息和删除购票信息。管理员首先创建自己的管理员账号，进入系统可以看到首页界面，里面有许多景区可以选择，并且可以看到对应的评论个数、景区的等级、前十的评分景区，然后可以在左侧导航栏中看到管理员中心，里面包括修改信息、修改密码。之后是管理中心，里面包括添加景点、景点管理页面和用户购票管理页，可以在景点管理界面看到对应的添加信息，添加完成点击确认，去往景点管理页，可以看到景点信息并且修改景点信息和删除景点信息。之后用户购票管理页可以看到所有游客的信息，并且能实现增删改查功能。

设计 Shine She 旅游系统功能，分两个角色创建功能模块，游客功能：修改个人信息、修改密码、购票、查看购票详情、修改购票信息、删除购票信息。管理员功能：修改管理员信息、修改密码、添加景点、查看景点详情、修改景点信息、删除景点信息、查看游客详情、修改购票信息、删除购票信息。

### 任务：

1. 应用所学知识和信息查询、文献检索等手段，分析系统需求，考虑社会、健康、安全、法律、文化及环境因素的影响。
2. 综合运用 Python 程序设计基础知识，选用合适的第三方库，设计和实现一个项目，包括概念设计、逻辑设计、功能模块设计与实现，并体现创新意识。
3. 分析项目对社会、健康、安全、法律以及文化的影响。
4. 针对项目解决方案，评价其计算效率、系统可靠性、可维护性和方便性措施，判断项目对人类、环境、社会效益的影响。
5. 撰写实训报告，制作答辩 ppt，参加答辩。

# 课程设计任务书

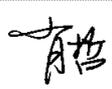
## 2. 对课程设计成果的要求（包括图表、实物等硬件要求）:

- (1) 课程设计报告、任务书要求同时提交电子版和纸质版材料
- (2) 课程设计报告中涉及到的所有表格都要求使用 Microsoft Visio 绘制,并要求从 Visio 直接复制进 Word,不得截图。
- (3) 所有的图例不要带底纹,图中文字建议使用 Visio 中的 14 号字;整体效果美观。
- (4) 所有表格请使用三线表,具体格式请参照课程设计报告模板。
- (5) 程序设计源代码请压缩后上传至指定邮箱或实训课堂。
- (6) 电子版资料请按以下方式命名:学号\_姓名\_班级\_课题名称.\*

## 3. 主要参考文献:

- [1] 赵广辉等. Python 程序设计基础[M]. 高等教育出版社, 2021, 5.
- [2] 嵩天等. Python 语言程序设计(第 2 版)[M]. 高等教育出版社, 2017, 2.
- [3] 明日科技, 零基础学 Python[M], 吉林大学出版社, 2018, 4.
- [4] 龚沛曾, 杨志强. Python 程序设计及应用[M]. 高等教育出版社, 2021, 10.
- [5] 叶惠仙, 游金水. Python 语言在大数据处理中的应用[J]. 网络安全技术与应用, 2021, (05): 51-54.
- [6] 汪洋, 姜新通. MVC 框架在 Python 与 Django 下的设计研究[J]. 电脑与信息技术, 2021, 29 (01): 55-57+63.
- [7] 胡钊. 基于 Python Flask 的温度数据可视化[J]. 电子世界, 2021, (08): 117-119.
- [8] 王珊, 张俊. 数据库系统概论(第 5 版)习题解析与实验指导[M]. 高等教育出版社, 2015, 7
- [9] 王雷斌. 计算机机房管理信息系统的分析与设计[J]. 通讯世界, 2020, 27(04) :89-90

## 4. 课程设计工作进度计划:

序号	起迄日期	工作内容
1	2024.5.22-2024.5.23	选题、需求分析
2	2024.5.23-2024.5.27	系统设计
3	2024.5.27-2024.5.29	系统实现、测试
4	2024.5.29-2024.5.30	撰写实训报告
5	2024.5.30-2024.5.31	答辩
主指导教师签名		日期:
		2024 年 6 月 2 日



湖南工业大学科技学院  
COLLEGE OF SCIENCE AND TECHNOLOGY HNUT

# 课程设计说明书

课程名称: Python 程序设计课程设计

设计题目: 基于 Django 框架 Shine She 旅游系统

专 业: 计算机科学与技术 班级: 计科 2106

学生姓名: 刘继友 学 号: 21412030655

指导教师: 肖 哲

湖南工业大学科技学院教务部 制

2024 年 6 月 2 日

# 目 录

1 引言（或绪论） .....	1
1.1 系统开发背景 .....	1
1.2 研究意义 .....	1
1.3 国内外研究现状 .....	1
1.4 关键技术分析 .....	2
1.4.1 Django .....	2
1.4.2 Sqlite3 .....	2
1.4.3 Navicat Premium 15 .....	3
1.5 系统的特色和创新之处 .....	3
1.6 本章小结 .....	3
2 需求分析 .....	4
2.1 可行性分析 .....	4
2.1.1 系统实施的影响分析 .....	4
2.1.2 技术可行性 .....	4
2.1.3 经济可行性 .....	4
2.1.4 操作可行性 .....	4
2.1.5 其他因素分析 .....	4
2.2 总体架构 .....	5
2.3 功能需求分析 .....	5
2.3.1 需求描述 .....	5
2.3.2 数据流图 .....	7
2.3.3 数据字典 .....	8
2.4 业务流程分析 .....	9
2.4.1 游客流程分析 .....	9
2.4.2 管理员流程分析 .....	9
2.5 系统的非功能性需求 .....	10
2.5.1 性能、容量 .....	10
2.5.2 可靠性 .....	10
2.5.3 可维护性 .....	10
2.5.4 安全 .....	10
2.6 本章小结 .....	10
3 系统设计 .....	11
3.1 系统设计 .....	11
3.1.1 功能模块图 .....	11
3.1.2 系统功能分析 .....	11
3.1.3 系统用例图 .....	12
3.2 数据库设计 .....	12
3.2.1 局部概念模式设计 .....	12
3.2.2 全局概念模式设计 .....	14
3.2.3 数据库设计 .....	14
3.3 本章小结 .....	15
4 系统设计与实现 .....	16

<b>4.1 系统相关功能的实现</b> .....	16
4.1.1 登录.....	16
4.1.2 注册.....	21
4.1.3 注销.....	22
4.1.4 个人中心功能.....	23
4.1.5 购票中心.....	25
4.1.6 管理员首页.....	30
4.1.7 管理员个人中心.....	31
<b>4.2 系统性能评价</b> .....	42
4.2.1 系统的易用性指标.....	42
4.2.2 系统的可扩展性指标.....	43
4.2.3 系统的健壮性指标.....	43
4.2.4 系统的安全性指标.....	43
4.2.5 系统中可能存在的隐患.....	43
<b>5 总 结</b> .....	44
<b>参考文献</b> .....	45

# 1 引言（或绪论）

## 1.1 系统开发背景

智慧旅游已超越单一的旅游范畴，进入到文体旅一体化融合发展阶段，这将有助于提升城市经济发展水平和城市发展形象，也为乡村振兴战略赋能。建设县域智慧旅游管控系统，利用物联网和大数据等科技手段，结合应急广播建设，实施对全县景区(点)、旅游安全、客流信息的数据分析、管控打造旅游热点区域，推动县域旅游创新提升。基于此，本文以湘西的旅游管理系统开发建设为例，对系统建设展开讨论。现如今经济快速发展和人们生活水平的提高，旅游已经成为越来越多人休闲娱乐的首选方式。旅游行业也随着旅游人数的增加而蓬勃发展。湘西因有着神秘的代号，因此旅游也是湘西发展的一个重要途径，然而由于经济稍显不足，从而导致旅游信息化管理与其他地区相比较为劣势。为了满足人们对湘西旅游信息的需求，为用户提供全面的旅游信息，帮助游客做出更好的旅行计划，同时为旅游企业带来更多的商业机会。且对于庞大的旅游数据进行信息化高效处理。因此，开发这样一个系统不仅具有实际应用价值，也具有重要的社会和经济意义。

## 1.2 研究意义

现代化的计算机信息管理系统在众多旅游服务业已得到广泛应用，满足了广大经营者的迫切需求，它不仅进一步完善了企业管理信息化的进程，更为企业管理走向科学化、规范化、网络化奠定了基础。根据大量用户的实际应用表明，采用计算机管理即可节省人力，又起到提高服务速度及准确率作用，同时还可提升当地旅游形象和信誉，提高顾客满意度，大大增强竞争能力和管理水平，综合效益非常明显。计算机辅助旅游管理，可以极大地增强管理者采集、处理信息的能力，从而有利于管理者及时决策，计算机系统能根据管理过程的变化情况，将原始数据、资料等进行加工保存，管理人员可以在解决具体问题需要信息资料时，随时进行检索查询，了解整个旅游管理系统的动态情况，进行动态管理，从而有效的处理用户旅游信息的管理工作，实现旅游管理信息管理的自动化，是与现代化旅游管理模式接轨的明智选择。同时系统可以提供全面的旅游信息，包括景点类型、热门景点等，帮助游客做出更好的旅行计划。旅游管理系统的研究对于提升旅游业的服务质量和管理水平具有重要意义。

## 1.3 国内外研究现状

在旅游业蓬勃发展的今天，越来越多的公司开始重视旅游业务系统的建设和应用。在国内，驴妈妈是一家拥有成熟旅游业务系统的知名公司。他们的系统涵盖了线上线下全渠道的门票、酒店、交通等资源，以及多样化的旅游产品。另外，携程也是旅游业务

系统的领军企业之一，他们的系统不仅服务于个人消费者，还向旅行社、企业客户提供定制化解决方案。

国际上，Sabre 是一家拥有丰富经验的旅游业务系统提供商。他们的系统涵盖了某航空公司、酒店、旅行社等不同类型的客户，可以满足不同规模和需求的企业。另外，Amadeus 也是一家在旅游业务系统领域拥有一席之地的公司，他们的系统涵盖了某航、酒店、租车等多个方面，可以为客户提供全面的解决方案。

## 1.4 关键技术分析

本系统数据存储使用本地的 MySQL 数据库并用 Python 程序设计语言来实现。通过使用 Sqlite3 相关技术来连接数据库，实现从数据库中查询、插入、更新、删除数据。系统页面采用 Django 和 web 前端的相关技术，用以提示操作、显示操作的结果，也可通过数据库可视化工具 Navicat 来查看操作的结果

### 1.4.1 Django

Django 是一个遵循 MVC 设计模式的 Python Web 开发框架，但它实际上采用的是 MTV (Model-Template-View) 的设计模式。在 Django 中，模型 (Model) 负责数据存取层，处理与数据相关的所有事务，如数据的存取、验证有效性、数据之间的关系等。模板 (Template) 则处理与表现相关的决定，即在页面或其他类型文档中如何显示数据。视图 (View) 是业务逻辑层，负责存取模型及调取恰当模板的相关逻辑，是模型与模板之间的桥梁。Django 的独特之处在于它将 MVC 中的视图进一步分解为 Django 视图和 Django 模板两个部分，分别决定“展现哪些数据”和“如何展现”。这种设计使得 Django 的模板可以根据需要随时替换，而不仅仅限制于内置的模板。至于 MVC 中的控制器部分，在 Django 中由框架的 URLconf 来实现，使用正则表达式匹配 URL，然后调用合适的 Python 函数。URLconf 对于 URL 的规则没有任何限制，可以设计成任意的 URL 风格。Django 的核心组件包括用于创建模型的对象关系映射、为最终用户设计的较好管理界面 URL 设计、设计者友好的模版语言以及缓存系统。它已经成为 web 开发者的首选框架之一，因其鼓励快速开发并遵循 MVC 设计模式而受到广泛欢迎。Django 的命名来源于比利时的爵士音乐家 Django Reinhardt，他是一个吉普赛人，主要以演奏吉它为主。由于 Django 在近年来的迅速发展，应用越来越广泛，被著名 IT 开发杂志 SD Times 评选为 2013 SD Times 100，位列“API、库和框架”分类第 6 位。

### 1.4.2 Sqlite3

SQLite3 是一款轻量级的、基于文件的嵌入式数据库，实现自包容、零配置、支持事务的 SQL 数据库引擎。与其他数据库管理系统不同，SQLite3 的安装和运行非常简单，在大多数情况下，只要确保 SQLite3 的二进制文件存在即可开始创建、连接和使用数据库。第一个 Alpha 版本诞生于 2000 年 5 月，直到今天已经成为最流行的嵌入式

数据库，包括 Google 在内的许多公司在其桌面软件中亦使用 SQLite3 存储用户数据，由此看来，其稳定性毋庸置疑。

### 1.4.3 Navicat Premium 15

Navicat 是一套数据库管理工具，专为简化数据库的管理及降低系统管理成本而设。Navicat 是以直觉化的图形用户界面而建的，可以安全和简单地创建、组织、访问并共用信息。Navicat Premium 是 Navicat 的产品成员之一，能简单并快速地在各种数据库系统间传输数据，或传输一份指定 SQL 格式及编码的纯文本文件。其他功能包括导入向导、导出向导、查询创建工具、报表创建工具、资料同步、备份、工作计划及更多。并且该数据库管理工具有许多应用途径，例如：跨平台兼容性：除了支持多种数据库，Navicat Premium 还可以与 Amazon RDS、Amazon Aurora、Amazon Redshift、Microsoft Azure、Oracle Cloud、MongoDB Atlas 以及中国的阿里云、腾讯云和华为云等云数据库服务兼容。这种云兼容性确保了用户可以无缝管理在云环境中运行的数据库。高效管理操作：Navicat Premium 提供了一系列的工具和功能，如对象设计器、SQL 编辑器、数据操作和模型图表等，这些功能可以帮助用户高效地创建、管理和优化数据库结构。它还支持数据的同步和分析，这对于需要维护多个数据库或进行数据分析的用户来说非常有用。安全性与协作：Navicat Premium 还注重安全性，提供了安全连接选项，确保数据传输的安全性。此外，它还支持协同合作，使得团队成员可以在同一项目上共享和同步工作进度。用户友好的界面：Navicat Premium 拥有直观的用户界面，使得即使是非专业的数据库用户也能快速上手并进行有效的数据库管理。它的设计旨在简化复杂的数据库操作，让用户能够专注于他们的核心业务。

## 1.5 系统的特色和创新之处

目前，现阶段市场上所呈现出的旅游系统，绝大部分是为大型的旅游企业进行设计的，优点是其管理功能普遍较为全面，但是其存在的不足也是非常的明显，例如许多功能并不适用于中小型的旅游企业。实际上，在当今国内市场中占据份额比例更大一块的却是中小型旅游企业，与之对应的却是适用于中小型企业的旅游系统数量非常少。因此，该系统主要是针对中小型旅游企业而开发的管理系统。并且该系统只针对湘西的一些特色地点进行景点的介绍、排名等。

## 1.6 本章小结

本章介绍了这次课题主要开发的背景和研究意义，并调查了该课题国内和国外的研究现状，概括了本系统所使用到的相关技术，以及本系统的特色创新之处。

## 2 需求分析

### 2.1 可行性分析

系统的可行性分析主要从系统实施的影响、技术可行性、经济可行性、操作可行性、法律可行性、风险评估与应对措施以及其他方面因素进行阐述和分析。

#### 2.1.1 系统实施的影响分析

对于旅游行业来说，旅游系统就是为了提高旅游行业的服务质量，提高旅游行业对于大数据的处理的效率，以及增加旅游行业经济收入，能更好的提高当地的影响力，带动周边经济，推动实现人民对美好生活的向往的目标。

#### 2.1.2 技术可行性

本次系统的开发用到的是 MySQL 数据库、Python 编程语言以及数据库可视化工具 Navicat Premium 15。使用这些软件开发一个中小型的旅游管理系统是绰绰有余的。首先，MySQL 体积小、速度快、总体拥有成本低，开源；支持多种操作系统；是开源数据库，提供的接口支持多种语言连接操作，而此次设计的是一个区域的旅游管理系统，里面包含的数据并不会包含太多，所以不会存在数据溢出的情况。其次，Python 语言自身所提供的各种控件、函数及方法完全可以帮助我们完成系统的开发，并且 Python 语言简洁，可以减少代码冗余，减少内存空间的使用。在实际的开发过程和应用中，数据库和应用程序均在同一计算机上运行。最后，Navicat Premium 15 可视化工具，可以在我们设计系统是提供更加简洁的数据添加功能，减少代码的输入，提高效率。

#### 2.1.3 经济可行性

对于旅游行业的发展来说，生活质量的提高使得越来越多的人群向往轻松的生活，这就使得旅游数据变得繁多，经济收入越来越多，而旅游管理系统这个软件的开发和维护只需要很少的开销，就能的到很大的回报，且能提高旅游数据处理的效率，创造更多的经济利益。

#### 2.1.4 操作可行性

本系统是系统采用模块化设计，便于维护和功能扩展。例如，Django 的应用结构支持按功能划分模块，易于管理和扩展。前端页面简洁移动，操作人员可根据界面提示进行操作，并且提供详细的用户手册、管理员手册和在线帮助文档，帮助用户和管理员快速上手。使用 Django 开发的旅游管理系统在操作可行性方面具有很高的潜力。系统设计应以用户体验为核心，提供简洁易用的界面和流畅的操作流程，同时注重系统的可维护性和安全性。通过良好的技术支持和培训，可以确保系统在实际运营中顺利运行，并为用户和管理员提供高效便捷的服务。

#### 2.1.5 其他因素分析

##### 1) 社会

本系统如果企业使用得当，不仅能为当地得到很好的经济收入而且可以很好的占领客户市场，从而占取更多的市场份额。同时在社会上能提高地区影响力，吸引更多人了解该地区，做到有效的文化数据输出。因此本系统对于社会主要就是经济发展方面的影响，和传统文化的传承。

## 2) 健康

本系统的使用者是企业的管理员或者旅游的游客，取代了以前手工录入的传统模式，完全由电脑端操作，大大减少了工作时间，用于日常的办公以及业务处理，对操作人员的健康不会产生不良影响或者威胁等。且减少了纸等公共资源的浪费，完全符合中国可持续发展的国情。

## 3) 法律

首先，本系统所管理的内容完全是符合社会主义要求的，不存在反国反党的言论或者信息；其次，本系统内的信息是十分健康的，不存在辱骂或者脏话的信息；最后，本系统不存在侵权问题。

## 4) 文化

本系统对企业管理文化有推动的作用，会逐渐覆盖传统的企业经营模式，推进了企业文化向着信息化发展，并且影响当地文化向全国输出，乃至向全世界输出。

## 5) 环境

本系统完全在电脑端操作，不会产生任何对环境有害的污染，且完全取代了以前手工录入的繁琐操作，减少了纸张的浪费的同时影响了树木的生长，符合中国可持续发展的国情，以及世界碳中和的理念。

## 2.2 总体架构

本系统的使用人有两种分别是：管理员和游客。主要的流程就是进入旅游系统后，对数据库的数据进行操作，修改数据库中的数据，然后页面的数据会根据数据库的操作变化而变化。本系统的总体架构图如表 2.1 所示

用户	管理员、游客
表现层	GUI
应用层	旅游管理系统
数据层	MySql 数据库

表 2.1 系统总体架构图

## 2.3 功能需求分析

### 2.3.1 需求描述

本系统有两类人使用，一种是管理员，另一种是游客。要求整个系统满足所有的需

求，且具有实际操作意义。

首先，登录界面：

1. 普通用户登录：有账号直接进入普通用户界面。
2. 注册账号：点击注册账号，进入注册界面，输入相关信息。
3. 管理员登录：有账号直接进入管理员界面。
4. 注册账号：点击注册账号，进入注册界面，输入相关信息。

其次，从管理员的角度出发：

1. 首页：可以看到景区的个数、评论个数、销量前十景点、评论前十景点。
2. 管理员中心：包括修改信息，修改密码。
  - (1) 修改信息：管理员可以修改自己的个人信息。
  - (2) 修改密码：管理员可以修改自己的密码。
3. 游客购票管理：包括查看游客购票的信息、修改游客购票信息，删除游客购票信息。
  - (1) 查看游客购票信息：管理员可有查看所有游客购票的信息。
  - (2) 修改游客购票信息：管理员可以修改游客的景区名称、预约的时间、真实的姓名、身份证号、联系方式。
  - (3) 删除游客购票信息：管理员可删除一条游客的预约信息。
4. 添加景点：管理员可以对首页的数据进行添加操作，添加的信息有景区名称、景点等级、景区地点、景区评分、价格、销量、详细地址、简介、详细简介等。
5. 景点管理：包括查看景点信息、修改景点信息、删除景点信息。
  - (1) 查看景点信息：管理员可以查看所有景点的信息。
  - (2) 修改景点信息：管理员可以修改景点的景区名称、景区名称、景点等级、景区地点、景区评分、价格、销量、详细地址、简介、详细简介等。
  - (3) 删除景点信息：管理员可删除一条景点的信息。

最后，从游客的角度出发：

1. 首页：可以看到景区的个数、评论个数、销量前十景点、评论前十景点。
2. 管理员中心：包括修改信息，修改密码。
  - (1) 修改信息：游客可以修改自己的个人信息。
  - (2) 修改密码：游客可以修改自己的密码。
3. 游客购票：游客可以根据首页提供的信息，进行购票，需输入景区名、预约日期、真实姓名、身份证号、联系方式。
4. 购票详情：包括查看购票信息、修改购票信息、删除购票信息。
  - (1) 查看购票信息：游客可以查看所有购票的信息。
  - (2) 修改购票信息：可以修改景区名称、预约的时间、真实的姓名、身份证号、

联系方式。

(3) 删除购票信息：可删除一条预约信息。

### 2.3.2 数据流图

顶层数据流图如图 2.2 所示

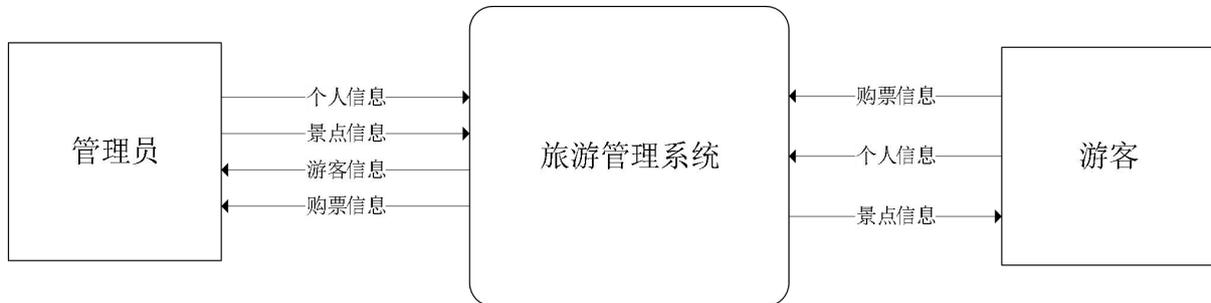


图 2.2 顶层数据流图

管理员一层数据流图如图 2.3 所示：

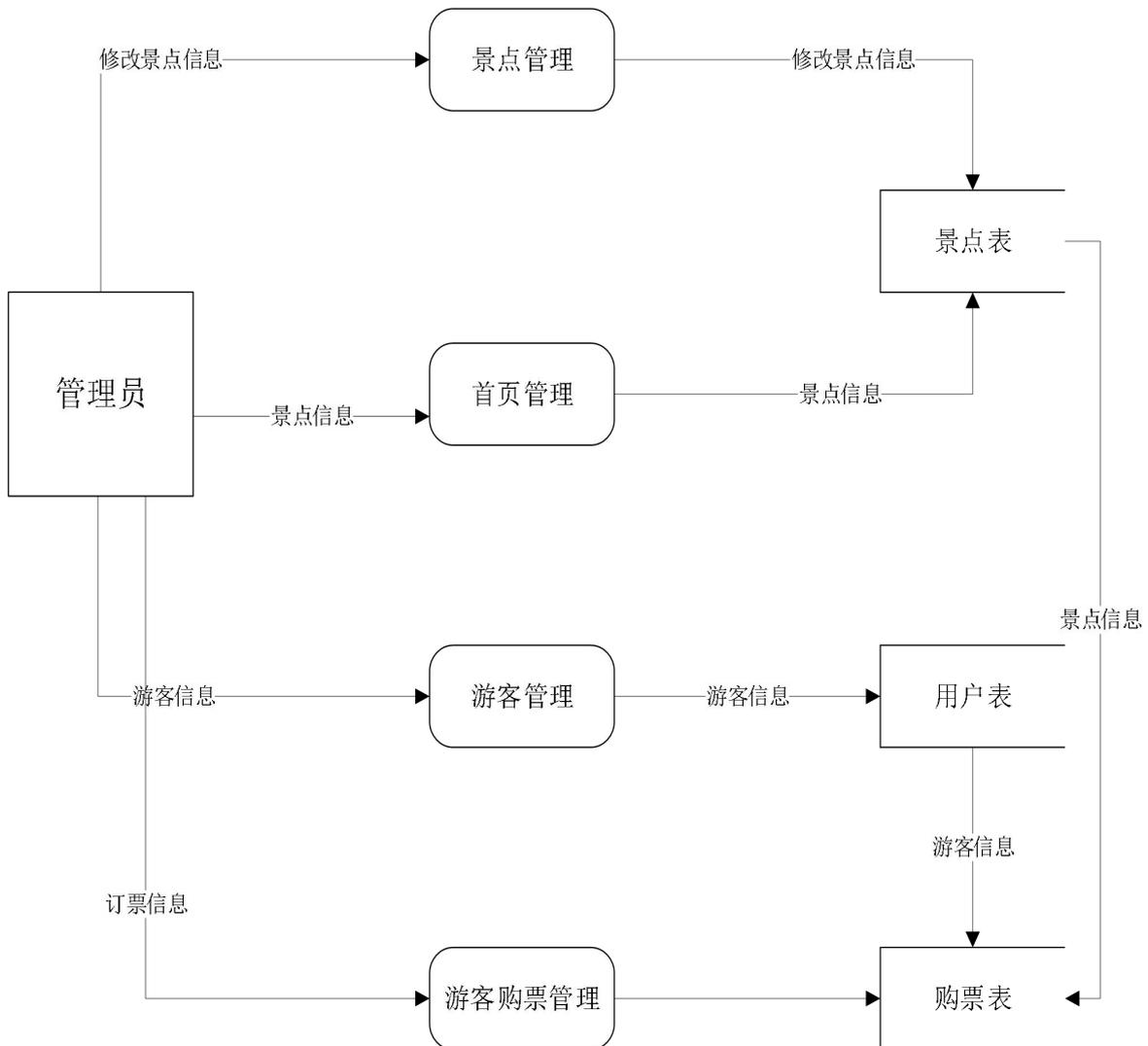


图 2.3 管理员一层数据流图

游客一层数据流图如图 2.4 所示：

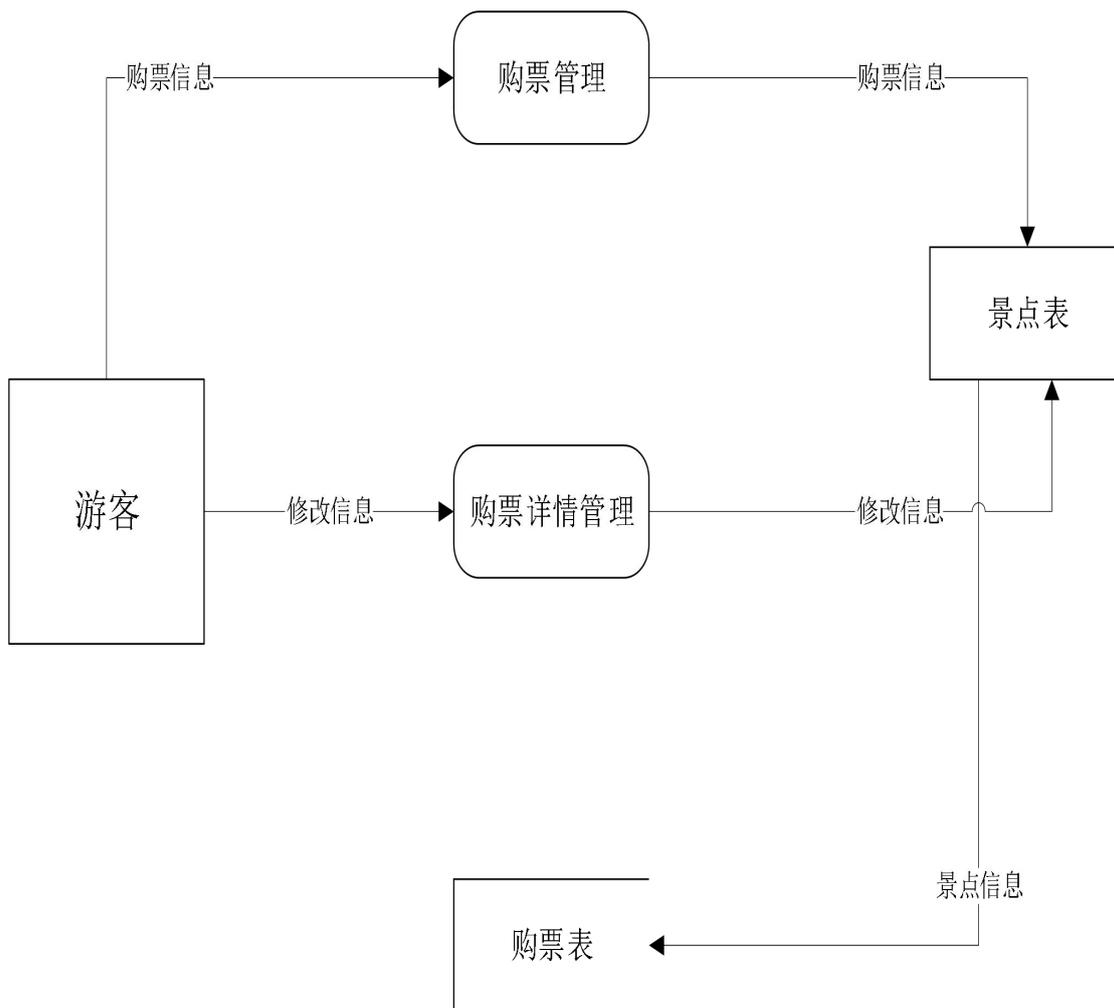


图 2.4 游客一层数据流

### 2.3.3 数据字典

景点信息 = 景点编号 + 景点名称 + 景点等级 + 景区地点 + 景区评分 + 价格 + 销量 + 详细地址 + 简介 + 详细简介

用户信息 = 用户编号 + 账号 + 密码 + 性别 + 地址 + 头像 + 简介

购票信息 = 购票编号 + 景区名称 + 预约时间 + 真实姓名 + 身份证号 + 联系方式

数据项列表如表 2.5 所示：

数据项名称	数据项含义	数据类型	长度
景点编号	唯一标识	int	0
景点名称	地点名称	varchar	255
景点等级	景区评级	varchar	255
景区地点	目的地	varchar	255
景区评分	评分	varchar	255
价格	门票价格	int	50

数据项名称	数据项含义	数据类型	长度
销量	销量	int	50
详细地址	地址	varchar	255
简介	简介	varchar	255
详细简介	详细简介	varchar	255
用户编号	唯一标识	int	0
账号	登录账号	varchar	255
密码	登录密码	int	255
性别	性别	varchar	10
地址	家庭地址	varchar	255
头像	账号头像	varchar	255
简介	账号简介	varchar	255
购票编号	唯一标识	int	0
景区名称	景区名称	varchar	255
预约时间	预约时间	date	0
真实姓名	姓名	varchar	255
身份证号	身份证号码	varchar	18
联系方式	手机号	int	11

表 2.5 数据项列表

## 2.4 业务流程分析

从游客、管理员两个角度分别分析业务流程

### 2.4.1 游客流程分析

游客首先创建自己的普通用户账号，进入系统可以看到首页界面，里面有许多景区可以选择，并且可以看到对应的评论个数、景区的等级、前十的评分景区，然后可以在左侧导航栏中看到个人中心，里面包括修改信息、修改密码。之后是购票中心，里面包括购票和购票详情页面，可以在购票界面看到对应的添加信息，添加完成点击确认，去往购票详情页，可以看到购票信息并且修改购票信息和删除购票信息。

### 2.4.2 管理员流程分析

管理员首先创建自己的管理员账号，进入系统可以看到首页界面，里面有许多景区可以选择，并且可以看到对应的评论个数、景区的等级、前十的评分景区，然后可以在左侧导航栏中看到管理员中心，里面包括修改信息、修改密码。之后是管理中心，里面包括添加景点、景点管理页面和用户购票管理页，可以在景点管理界面看到对应的添加信息，添加完成点击确认，去往景点管理页，可以看到景点信息并且修改景点信息和删除景点信息。之后用户购票管理页可以看到所有游客的信息，并且能实现增删改查功能。

## 2.5 系统的非功能性需求

分析系统的非功能性需求进，主要从系统性能容量、系统维护、安全等方面考虑。

### 2.5.1 性能、容量

本系统开发使用的软件以及技术，都是以简便、快速为目的，所以性能方面是不错的，反应速度也较快。系统使用的 MySQL 数据库，容量足以承受中小型旅游企业的信息量需求。

### 2.5.2 可靠性

系统开发使用的技术和软件都是被普遍使用的，不会轻易崩溃导致系统无法运行。同时，对数据的管理是严格的，因为会设定职位的特定权限，职位以外的数据都不可以操作。并且用户与用户之间的数据不能进行操作。

### 2.5.3 可维护性

本系统的开发采用的是 B/S 架构，浏览器端是不需要进行维护的，就省去了很多的麻烦，只需要专门的对网站的内容信息进行维护即可。其使用的软件为 Python、MySql 这些市面上常见的软件，并且是开源的软件，维护起来也不会产生相应的费用。

### 2.5.4 安全

系统使用了拦截器，数据库中没有信息的人是无法登录进系统的，离职的员工也无法登录，企业员工拥有的操作权限也不同，不会有随意登录导致信息泄露的风险。系统确保只有经过适当身份验证的用户才能访问系统资源，并且他们只能访问他们被授权访问的资源。系统实施访问控制机制，如访问控制列表（ACLs）和角色基础的访问控制（RBAC），以限制对系统和数据的访问。系统对敏感数据进行加密，以防止未授权的披露，特别是在数据传输和存储过程中。本系统使用防火墙、入侵检测系统（IDS）、入侵预防系统（IPS）和其他网络安全工具来保护系统免受网络攻击。

## 2.6 本章小结

本章主要是对系统的功能性中使用数据流图分别展现管理员和游客功能性分析，并且对本系统中的景点信息、用户信息、购票信息的这些数据进行数据列表的编写再对其进行分析。对管理员和游客进行业务流程分析阐述了两个角色之间的功能，以及非功能性需求的性能、容量、可靠性、可维护性、安全性分析。

### 3 系统设计

#### 3.1 系统设计

##### 3.1.1 功能模块图

功能过模块如图 3.1 所示：

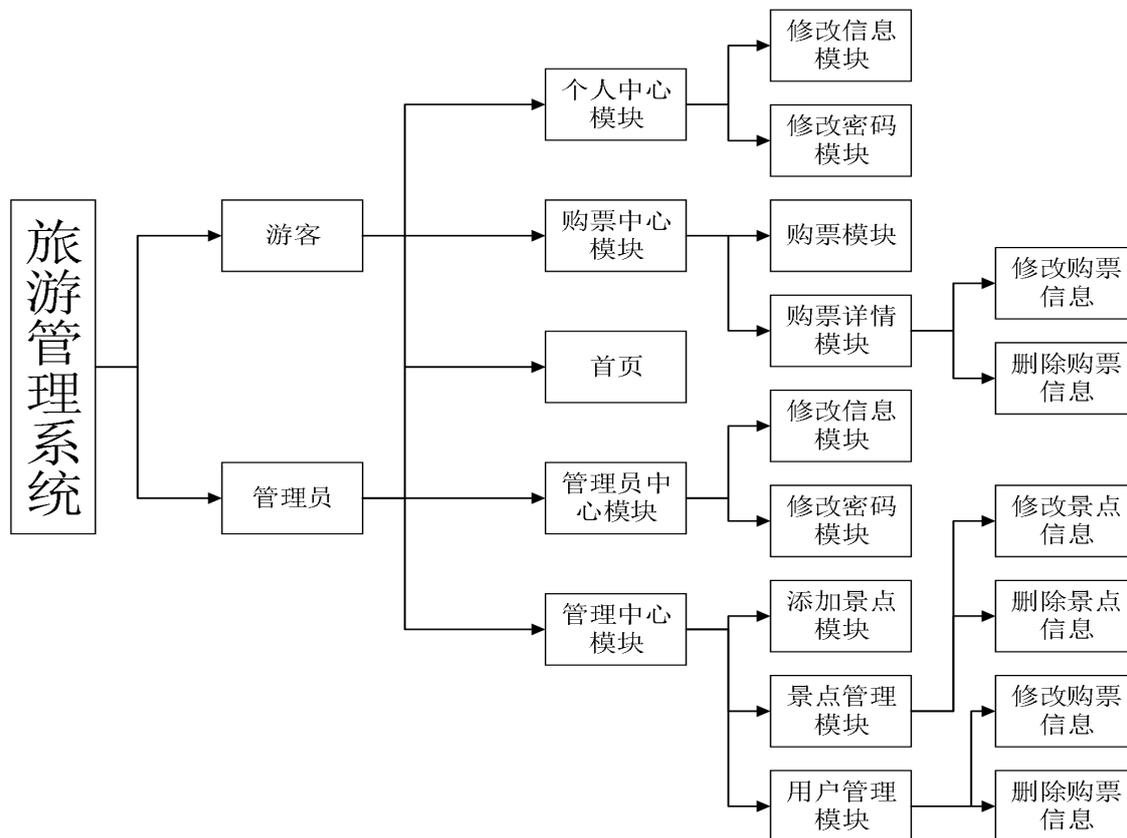


图 3.1 功能模块图

##### 3.1.2 系统功能分析

游客：

游客首先创建自己的普通用户账号，进入系统可以看到首页界面，里面有许多景区可以选择，并且可以看到对应的评论个数、景区的等级、前十的评分景区，然后可以在左侧导航栏中看到个人中心，里面包括修改信息、修改密码。之后是购票中心，里面包括购票和购票详情页面，可以在购票界面看到对应的添加信息，添加完成点击确认，去往购票详情页，可以看到购票信息并且修改购票信息和删除购票信息。

管理员：

管理员首先创建自己的管理员账号，进入系统可以看到首页界面，里面有许多景区可以选择，并且可以看到对应的评论个数、景区的等级、前十的评分景区，然后可以在左侧导航栏中看到管理员中心，里面包括修改信息、修改密码。之后是管理中心，里面包括添加景点、景点管理页面和用户购票管理页，可以在景点管理界面看到对应的添加

信息，添加完成点击确认，去往景点管理页，可以看到景点信息并且修改景点信息和删除景点信息。之后用户购票管理页可以看到所有游客的信息，并且能实现增删改查功能。

### 3.1.3 系统用例图

系统用例图如图 3.2 所示：

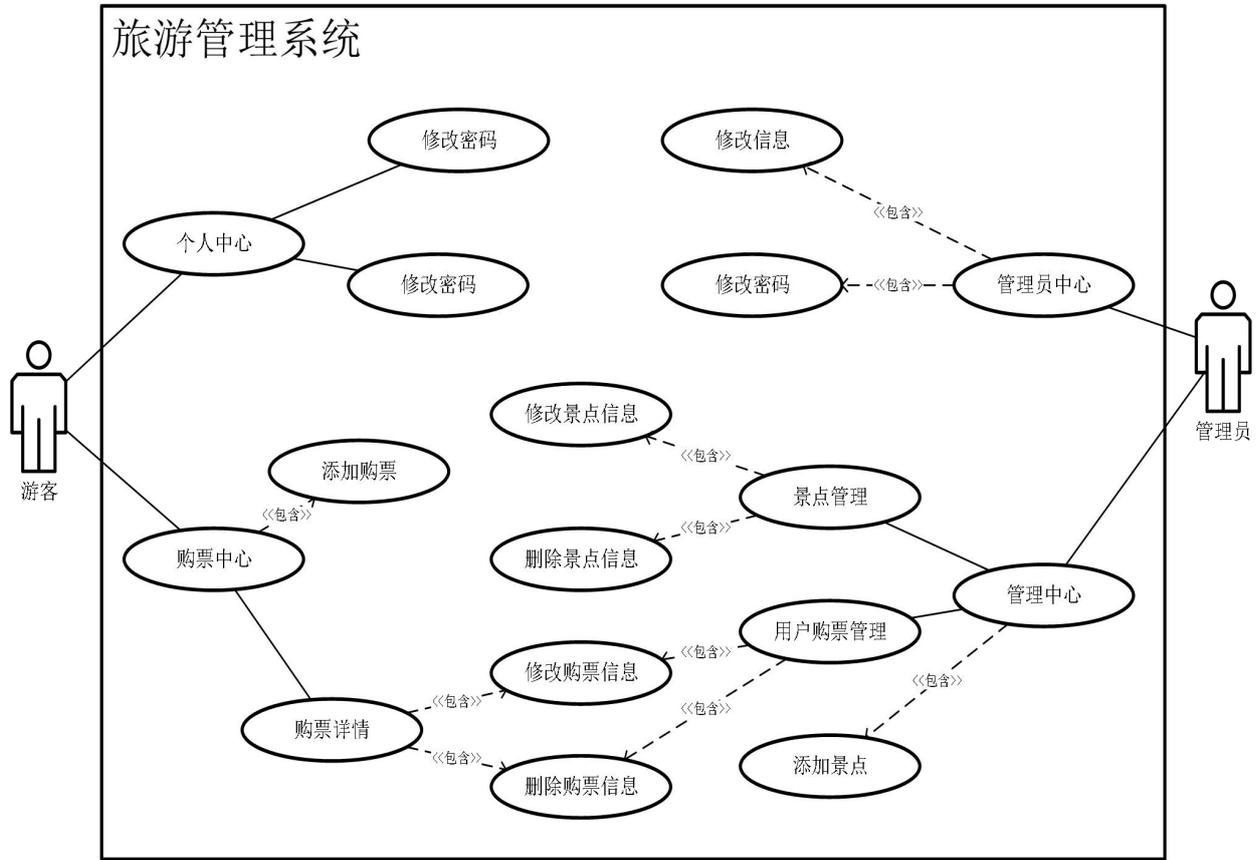


图 3.2 系统用例图

## 3.2 数据库设计

### 3.2.1 局部概念模式设计

景点实体的属性图如图 3.3 所示：

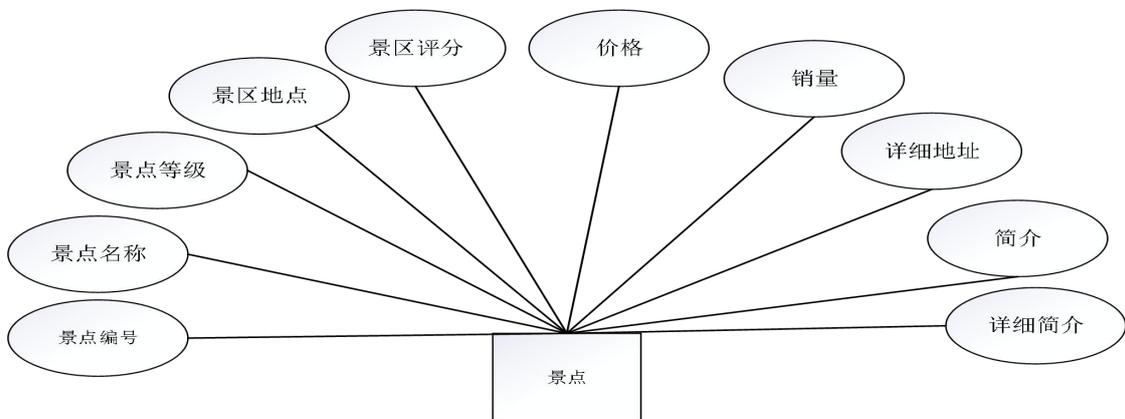


图 3.3 景点实体的属性图

用户的属性有用户编号、账号、密码、性别、地址、头像、简介。用户实体的属性图如图 3.4 所示：

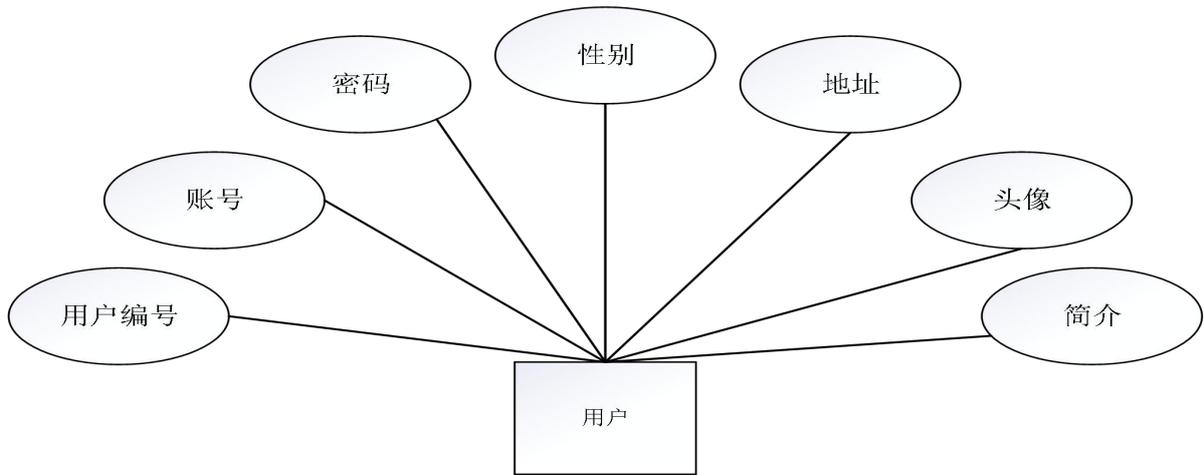


图 3.4 用户实体的属性图

购票信息的属性有购票编号、景区名称、预约时间、真实姓名、身份证号。可联系方式。购票实体的属性图如图 3.5 所示：

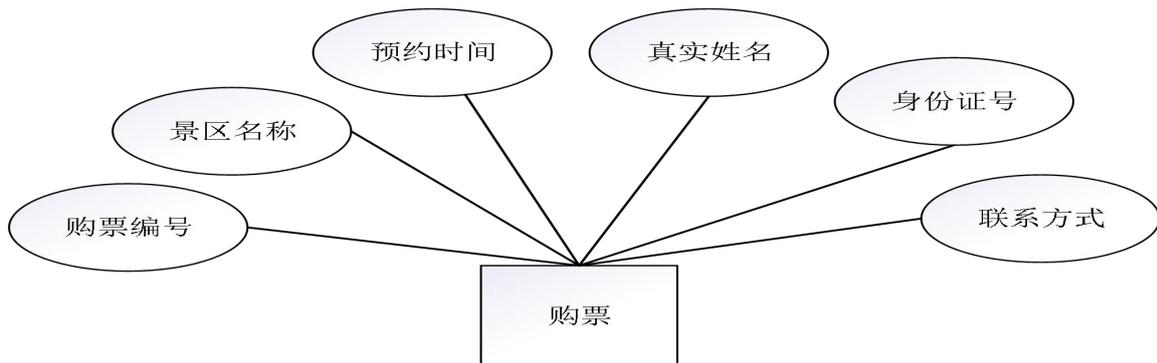


图 3.5 购票实体的属性图

管理员的属性有管理员编号、账号、密码、性别、地址、头像、简介。管理员实体的属性图如图 3.6 所示：

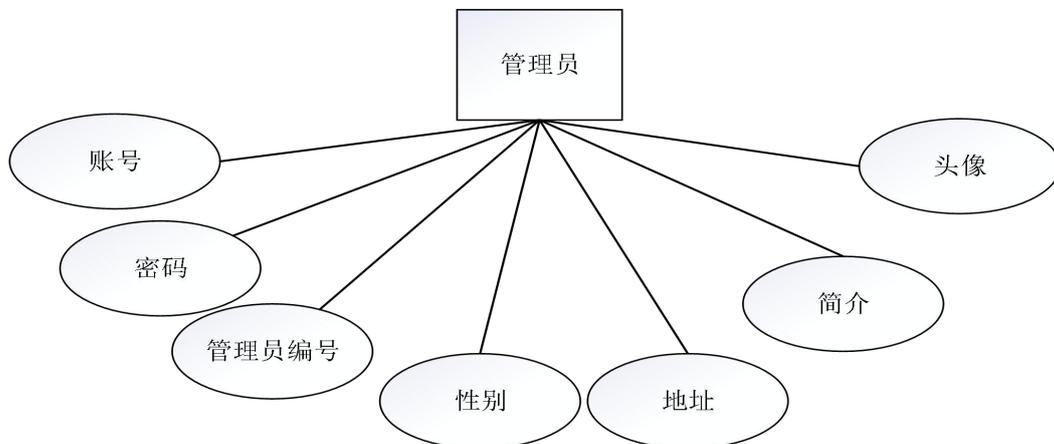


图 3.6 管理员实体的属性图

### 3.2.2 全局概念模式设计

全局概念模式图，将三个实体属性图综合在一起构成全局概念图，如下图 3.7 全局概念图所示：

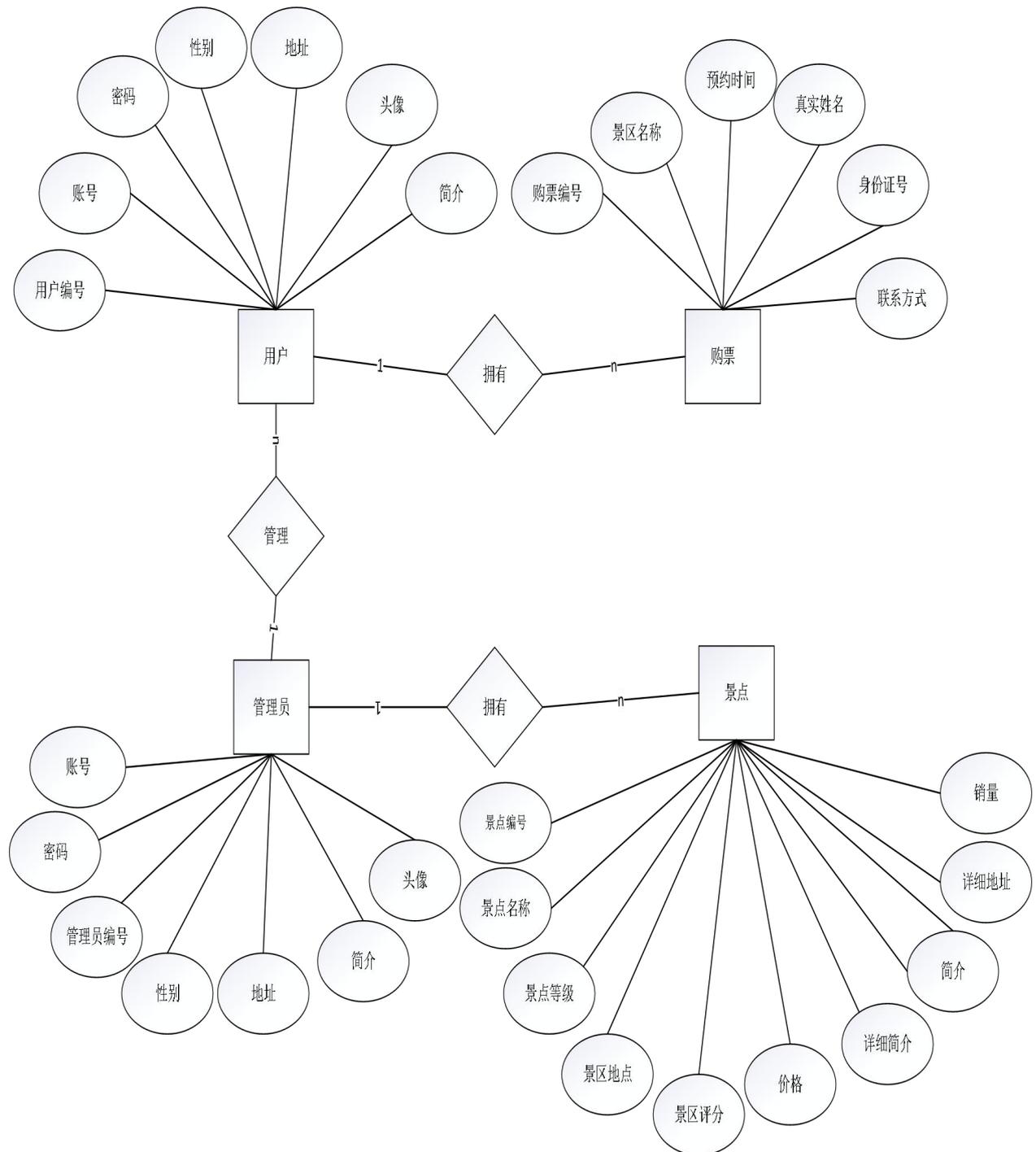


图 3.7 全局概念图

### 3.2.3 数据库设计

景点信息数据表包含景点编号、景点名称、景点等级、景区地点、景区评分、价格、销量、详细地址、简介、详细简介。景点属性表如下 3.8 所示所示

列名	数据类型	长度	允许空	字段说明
----	------	----	-----	------

景点编号	int	0	否	唯一标识
景点名称	varchar	255	否	地点名称
景点等级	varchar	255	否	景区评级
景区地点	varchar	255	是	目的地
景区评分	varchar	255	是	评分
价格	int	50	否	门票价格
销量	int	50	否	销量
详细地址	varchar	255	是	地址
简介	varchar	255	是	简介
详细简介	varchar	255	是	详细简介

表 3.8 景点属性表

用户信息数据表包含用户编号、账号、密码、性别、地址、头像、简介。用户属性表如下 3.9 所示所示

列名	数据类型	长度	允许空	字段说明
用户编号	int	0	否	唯一标识
账号	varchar	255	否	登录账号
密码	int	255	否	登录密码
性别	varchar	10	否	性别
地址	varchar	255	是	家庭地址
头像	varchar	255	是	账号头像
简介	varchar	255	是	账号简介

表 3.9 用户属性表

购票信息数据表包含购票编号、景区名称、预约时间、真实姓名、身份证号、联系方式。用户属性表如下 3.10 所示所示

列名	数据类型	长度	允许空	字段说明
购票编号	int	0	否	唯一标识
景区名称	varchar	255	否	景区名称
预约时间	date	0	否	预约时间
真实姓名	varchar	255	否	姓名
身份证号	varchar	18	否	身份证号码
联系方式	int	11	否	手机号

表 3.10 购票属性表

### 3.3 本章小结

本章主要分析系统的功能模块，将系统中的用户信息、购票信息、景点信息的实体属性分出来之后分别绘制 E-R 图出来，最后综合成全局 E-R 图，然后根据 E-R 图，设计用户数据库、景点数据库、购票数据库，并做出相应的数据库表。

## 4 系统设计与实现

### 4.1 系统相关功能的实现

#### 4.1.1 登录

首先进入登录界面选择登录成分进行账号和密码的输入，具体的登录流程图如图 4.1 所示。

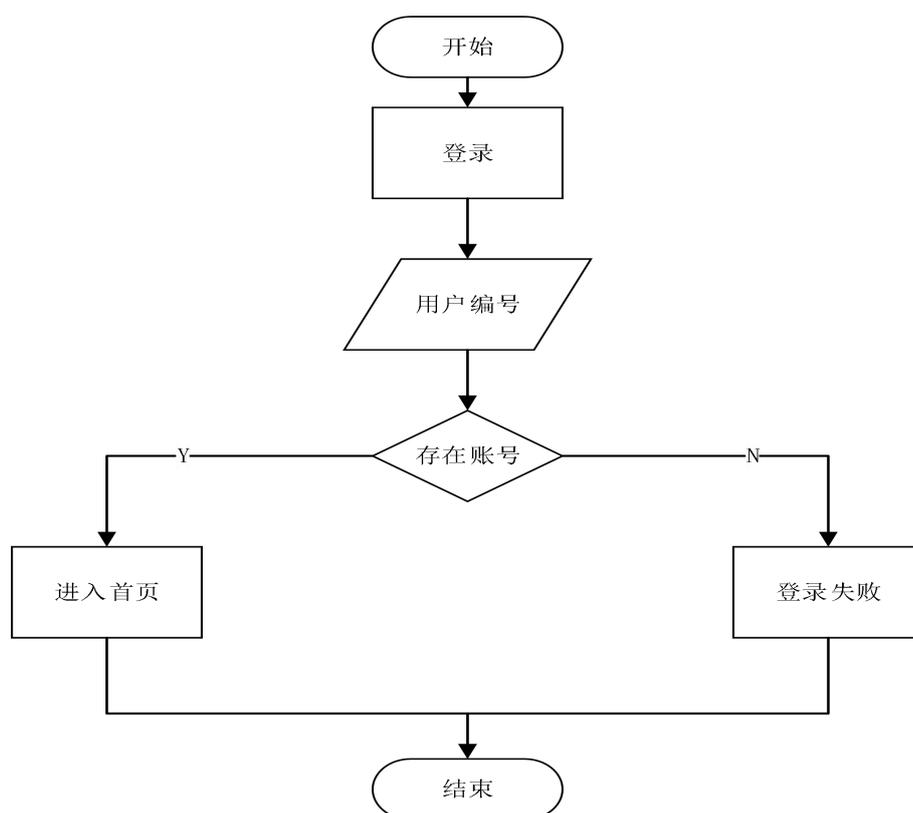


图 4.1 登录流程图

如果已经拥有账号登录就会成功进入首页。以下展示登录失败页面图 4.2，登录页面图 4.3，登录成功首页图 4.4。



图 4.2 登录失败图



图 4.3 登录页面图

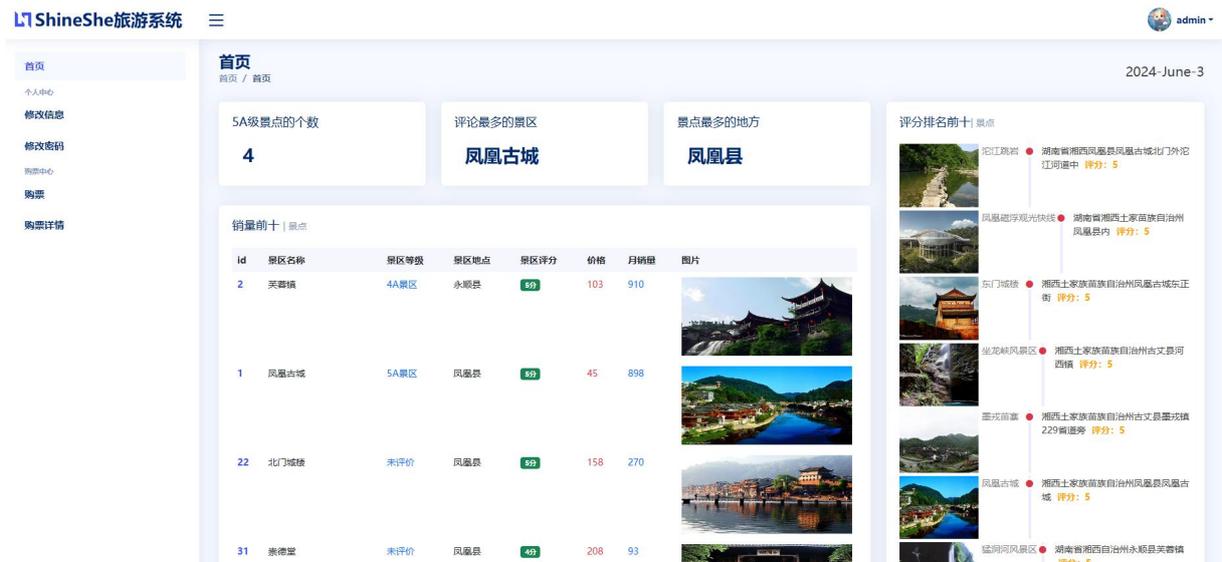


图 4.4 登录成功首页图

登录模块的相关后台代码

```
def login(request):
    if request.method == 'GET':
        return render(request, 'login.html')
    else:
        username = request.POST.get('username')
        password = request.POST.get('password')
        cd = username
        zhlei = request.POST.get('zhlei')
```

```

data = {cd:zhlei}
if zhlei == "普通用户":
    try:
        User.objects.get(username=username, password=password) #get
查询, 如果未存在就执行 except
        request.session['username'] = username
        return redirect('/SS01/index')
    except:
        return ErrorResponse.errorResponse(request, '账号或密码错误!
') #utils 中的模块, 调用 errorMsg 函数
    else:
        try:
            User.objects.get(username=username, password=password) #
get 查询, 如果未存在就执行 except
            request.session['username'] = username
            return redirect('/SS01/Mindex')
        except:
            return ErrorResponse.errorResponse(request, '账号或密码错误!
') # utils 中的模块, 调用 errorMsg 函数

```

首页模块的相关后台代码:

```

from SS01.utils import getPublicData
import random
import time
travelMapData = getPublicData.getAllTravelInfoMapData()
def getIndexTagData(): #获得数据库中的数据
    A5Len = 0
    commentsLenMax = 0
    commentsLenTitle = ''
    provienceDic = {}
    for travel in travelMapData:
        if travel.level == '5A 景区':
            A5Len += 1
        if int(travel.commentsLen) > commentsLenMax:

```

```

        commentsLenMax = int(travel.commentsLen)
        commentsLenTitle = travel.title
    if provienceDic.get(travel.province,-1) == -1:
        provienceDic[travel.province] = 1
    else:
        provienceDic[travel.province] += 1
    provienceDicSort = list(sorted(proviencDic.items(),key=lambda
x:x[1],reverse=True))[0][0]
    return A5Len,commentsLenTitle,proviencDicSort
def getAnthorData(): #获得数据库中评分数据
    scoreTop10 = []
    for travel in travelMapData:
        if travel.score == '5':
            scoreTop10.append(travel)
    maxLen = len(scoreTop10)
    scoreTop10Data = []
    for i in range(10):
        randomIndex = random.randint(0, maxLen - 1)
        scoreTop10Data.append(scoreTop10[randomIndex])
    saleCountTop10 = list(sorted(travelMapData, key=lambda
x:int(x.saleCount),reverse=True))[:10]#对销量前十进行排序
    return scoreTop10Data,saleCountTop10
def getNowTime(): #时间
    timeFormat = time.localtime()
    year = timeFormat.tm_year
    mon = timeFormat.tm_mon
    day = timeFormat.tm_mday
    return year,mon,day
def getGeoData():
    dataDic = {}
    for i in travelMapData:
        for j in getPublicData.cityList:
            for city in j['city']:
                if city.find(i.province) != -1:

```

```

        if dataDic.get(j['province'],-1) == -1:
            dataDic[j['province']] = 1
        else:
            dataDic[j['province']] += 1;resultData = []
for key,value in dataDic.items():
    resultData.append({
        'name':key,
        'value':value,
    })
return resultData
def index(request):
    username = request.session.get('username') #在后台有一个 username 的值,
    只有登录以后才会跳转到首页页面
    userInfo = User.objects.get(username=username)
    A5Len,          commentsLenTitle,          provienceDicSort          =
getIndexData.getIndexTagData()
scoreTop10Data,saleCountTop10 = getIndexData.getAnthorData()
year,mon,day = getIndexData.getNowTime()
geoData = getIndexData.getGeoData()
return render(request,'index.html',{
    'userInfo': userInfo,
    'A5Len':A5Len,
    'commentsLenTitle': commentsLenTitle,
    'provienceDicSort': provienceDicSort,
    'scoreTop10Data':scoreTop10Data,
    'nowTime':{
        'year': year,
        'mon': getPublicData.monthList[mon-1],
        'day': day,
    },
    'geoData':geoData,
    'saleCountTop10':saleCountTop10,
})
from SS01.models import TravelInfo,User

```

```

import json
import time
def getAllTravelInfoMapData():
    def map_fn(item):
        item.img_list = json.loads(item.img_list)
        item.comments = json.loads(item.comments)
        return item
    travelList = TravelInfo.objects.all()
    travelListMap = list(map(map_fn, travelList))
    return travelListMap
def getAllUsersData():
    return User.objects.all()

```

用 django 中的模型 #直接

#### 4.1.2 注册

当没有这些账号的时候点击创建账号，跳转注册界面，注册时也要选择注册成分，不然默认选择普通用户，注册流程图如下 4.5 所示

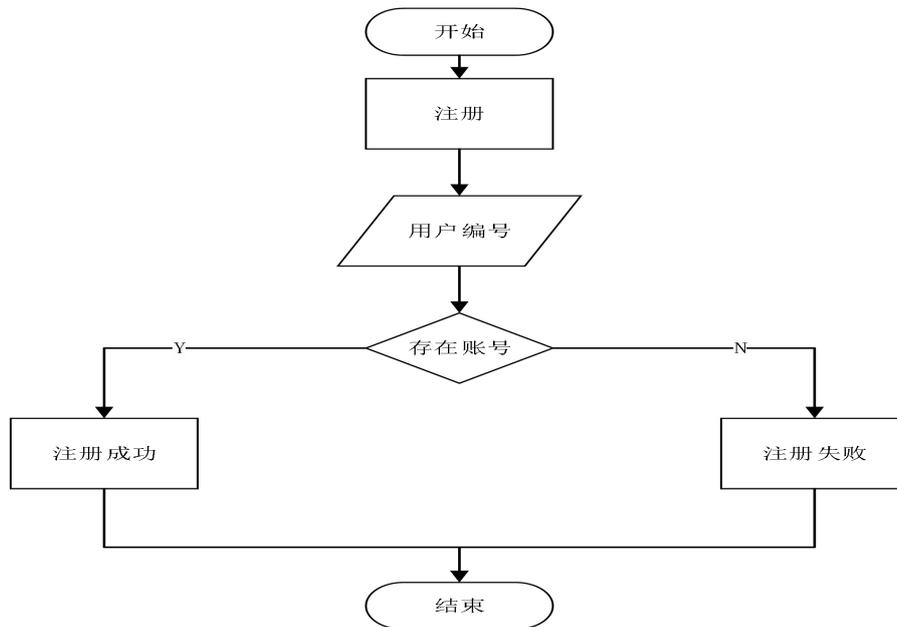


图 4.5 注册流程图

注册成功则返回登录界面进行登录操作，否则重新注册，实现的模块代码如下：

```

def register(request):
    if request.method == 'GET':
        return render(request, 'register.html')
    else:

```

```

zhlei = request.POST.get('zhlei')
username = request.POST.get('username')
password = request.POST.get('password')
confirmPassword = request.POST.get('confirmPassword')
try:
    User.objects.get(username=username) #get 查询, 如果未存在就执行
except 创建用户
except:
    if not username or not password or not confirmPassword or not zhlei:
        return HttpResponseRedirect('不允许为空!')
    if password != confirmPassword:
        return HttpResponseRedirect('两次密码不一致!')

    User.objects.create(username=username, password=password, Zhlei=z
hlei)#创建一个用户
    return redirect('/SS01/login')
return HttpResponseRedirect(request, '该账号已存在')

```

#### 4.1.3 注销

进入首页后如果要退出登录可以点击左上角头像或文字或小三角可看见注销账号的文字点击就退出登录, 返回到登录界面, 注销界面如图 4.6 所示



图 4.6 注销界面

实现的代码模块如下所示:

```

def logOut(request):
    request.session.clear()
    return redirect('/SS01/login')

```

#### 4.1.4 个人中心功能

普通用户进入首页可以在个人中心修改信息和修改密码，修改信息流程图 4.7 所示，修改密码流程图 4.8 所示，如下：

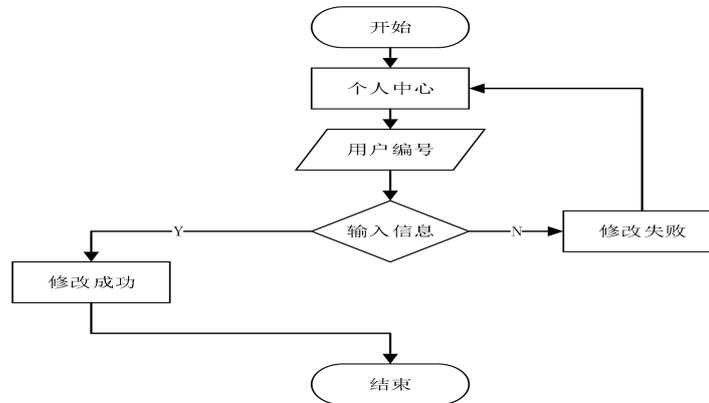


图 4.7 修改信息流程图

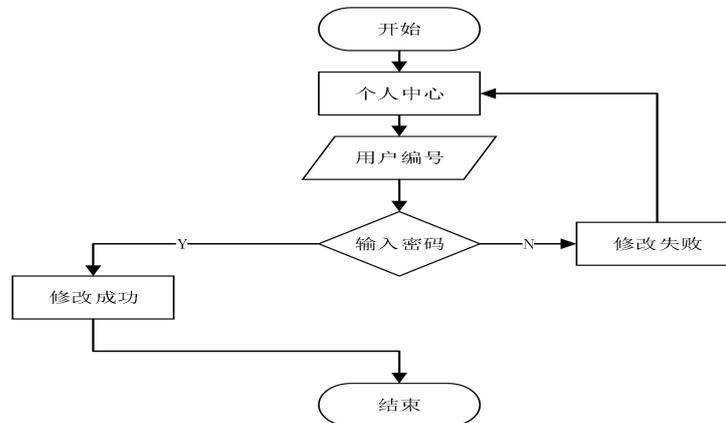


图 4.8 修改密码流程图

点击左侧导航条“个人中心”->“修改信息”或者“个人中心”->“修改密码”就可以进入相应的界面。如图 4.9 修改信息图、图 4.10 修改密码图



如图 4.9 修改信息图



图 4.10 修改密码图

修改信息和修改密码相关代码段如下：

```
def changeSelfInfo(username, formData, file):
    user = User.objects.get(username=username)
    user.address = formData['address']
    user.sex = formData['sex']
    if formData['textarea']:
        user.textarea = formData['textarea']
    if file.get('avatar') != None:
        user.avatar = file.get('avatar')
    user.save()

def getChangePassword(userInfo, passwordInfo):
    oldPwd = passwordInfo['oldPassword']
    newPwd = passwordInfo['newPassword']
    newPwdConfirm = passwordInfo['newPasswordConfirm']
    user = User.objects.get(username=userInfo.username)
    if oldPwd != userInfo.password: return '初始密码错误'
    if newPwd != newPwdConfirm: return '两次密码不一致'
    user.password = newPwd
    user.save()

def changeSelfInfo(request):
    username = request.session.get('username')
    userInfo = User.objects.get(username=username)
    year, mon, day = getIndexData.getNowTime()
    if request.method == 'POST':
```

```

getChangeSelfInfo.changeSelfInfo(username, request.POST, request.FILES)
userInfo = User.objects.get(username=username)
return render(request, 'changeSelfInfo.html', {
    'userInfo': userInfo,
    'nowTime': {
        'year': year,
        'mon': getPublicData.monthList[mon - 1],
        'day': day,
    },
})

```

```

def changePassword(request):
    username = request.session.get('username')
    userInfo = User.objects.get(username=username)
    year, mon, day = getIndexData.getNowTime()
    if request.method == 'POST':
        res = getChangeSelfInfo.getChangePassword(userInfo, request.POST)
        if res != None:
            return ErrorResponse.errorResponse(request, res)
    return render(request, 'changePassword.html', {
        'userInfo': userInfo,
        'nowTime': {
            'year': year,
            'mon': getPublicData.monthList[mon - 1],
            'day': day,
        },
    })

```

#### 4.1.5 购票中心

购票中心分为购票和购票详情界面，购票则是添加游客的购票信息，购票详情则是可以修改游客的购票信息和删除购票信息，图 4.11 添加游客流程图，图 4.12 购票详情流程图如下所示：

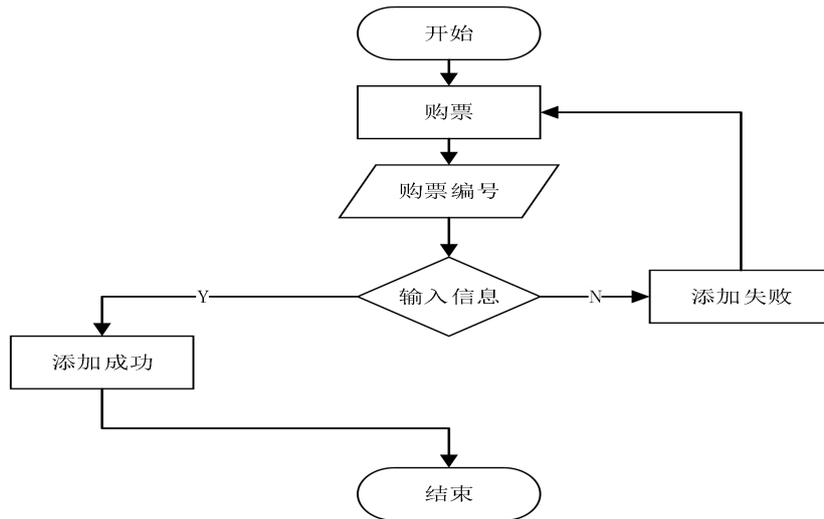


图 4.11 添加游客流程图

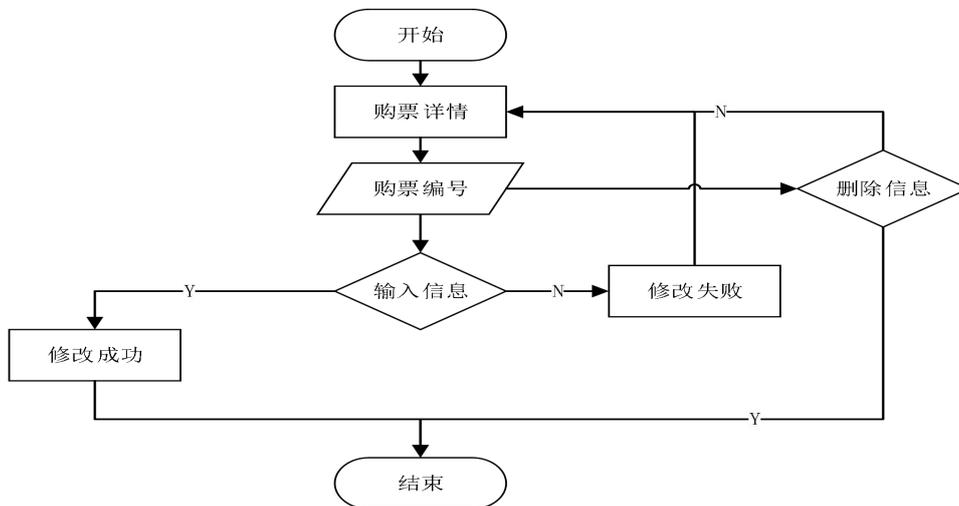


图 4.12 购票详情界面流程图

点击左侧导航条“购票中心”->“购票”或者“购票中心”->“购票详情”就可以进入相应的界面购票详情有修改和删除操作。如图 4.13 购票图、图 4.14 购票详情图、图 4.15 购票修改图

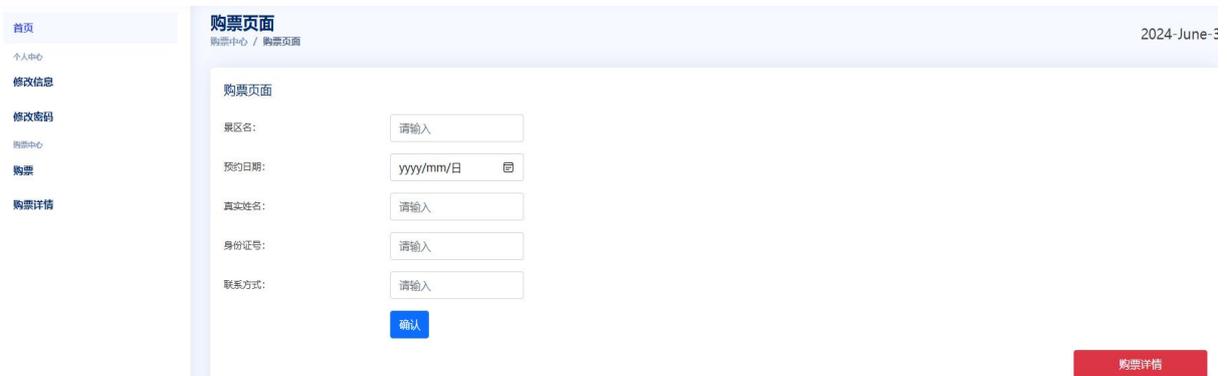


图 4.13 购票图



图 4.14 购票详情图

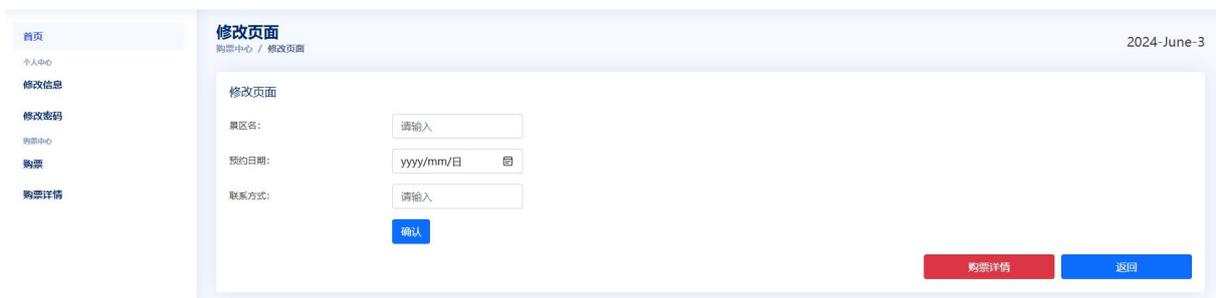


图 4.15 购票修改图

购票中心中的购票和购票详情页面代码模块如下所示:

购票代码段:

```
def saleTicket(request):
    username = request.session.get('username')
    userInfo = User.objects.get(username=username)
    year, mon, day = getIndexData.getNowTime()
    if request.method == 'POST':
        title = request.POST.get('title')
        cday = request.POST.get('cday')
        name = request.POST.get('name')
        idcard = request.POST.get('idcard')
        connection = request.POST.get('connection')
        if title == '':
            return HttpResponse('景区名不能为空! 请返回重新填写!')
        if cday == '':
            return HttpResponse('预约日期不能为空! 请返回重新填写!')
```

```

if name == '':
    return HttpResponse('您的姓名不能为空! 请返回重新填写!')
if idcard == '':
    return HttpResponse('身份证号不能为空! 请返回重新填写!')
if connection == '':
    return HttpResponse('联系方式不能为空! 请返回重新填写!')
SaleTicketDetail.objects.create(
    title=title,
    day=cday,
    name=name,
    Idcard=idcard,
    connection=connection,
) # 创建数据
return render(request, 'saleTicket.html', {
    'userInfo': userInfo,
    'nowTime': {
        'year': year,
        'mon': getPublicData.monthList[mon - 1],
        'day': day,
    },
})

```

购票详情代码段:

```

def saleTicketInfo(request):
    username = request.session.get('username')
    userInfo = User.objects.get(username=username)
    saleTicketInfoData = SaleTicketDetail.objects.all()
    year, mon, day = getIndexData.getNowTime()
    return render(request, 'saleTicketInfo.html', {
        'userInfo': userInfo,
        'saleTicketInfoData':saleTicketInfoData,
        'nowTime': {
            'year': year,
            'mon': getPublicData.monthList[mon - 1],
            'day': day,
        }
    })

```

```
    },  
  })
```

购票详情修改代码段:

```
def ticketFox(request):  
    username = request.session.get('username')  
    userInfo = User.objects.get(username=username)  
    year, mon, day = getIndexData.getNowTime()  
    nid = request.GET.get('nid')  
    if request.method == 'POST':  
        titlefox = request.POST.get('titleFox')  
        cdayfox = request.POST.get('cdayFox')  
        connectionfox = request.POST.get('connectionFox')  
  
        if titlefox == '':  
            return HttpResponse('景区名不能为空! 请返回重新填写!')  
        elif cdayfox == '':  
            return HttpResponse('预约时间不能为空! 请返回重新填写!')  
        elif connectionfox == '':  
            return HttpResponse('联系方式不能为空! 请返回重新填写!')  
        else:  
            SaleTicketDetail.objects.filter(id=nid).update(  
                title=titlefox,  
                day=cdayfox,  
  
                connection=connectionfox  
            )  
            return redirect("/SS01/saleTicketInfo/")  
    return render(request, 'ticketFox.html', {  
        'userInfo': userInfo,  
        'nowTime': {  
            'year': year,  
            'mon': getPublicData.monthList[mon - 1],  
            'day': day,  
        },  
    },
```

```
})
```

购票详情删除代码段：

```
def saleTicketInfo_delete(request):  
    nid = request.GET.get('nid')  
    SaleTicketDetail.objects.filter(id=nid).delete()  
    return redirect('../')
```

#### 4.1.6 管理员首页

图 4.16 管理员首页图如下所示



图 4.16 管理员首页图

管理员首页代码段展示：

```
def Mindex(request):  
    username = request.session.get('username') # 在后台有一个 username 的值，  
    只有登录以后才会跳转到首页页面  
    userInfo = User.objects.get(username=username)  
    A5Len, commentsLenTitle, provienceDicSort =  
getIndexData.getIndexTagData()  
    scoreTop10Data, saleCountTop10 = getIndexData.getAnthorData()  
    year, mon, day = getIndexData.getNowTime()  
    geoData = getIndexData.getGeoData()  
    return render(request, 'Mindex.html', {  
        'userInfo': userInfo,  
        'A5Len': A5Len,  
        'commentsLenTitle': commentsLenTitle,  
        'proviencedicSort': provienceDicSort,  
        'scoreTop10Data': scoreTop10Data,  
        'nowTime': {  
            'year': year,
```

```

    'mon': getPublicData.monthList[mon - 1],
    'day': day,
  },
  'geoData': geoData,
  'saleCountTop10': saleCountTop10,
})

```

#### 4.1.7 管理员个人中心

管理员进入首页可以在管理员中心修改信息和修改密码，修改信息流程图 4.17 所示，修改密码流程图 4.18 所示，如下：

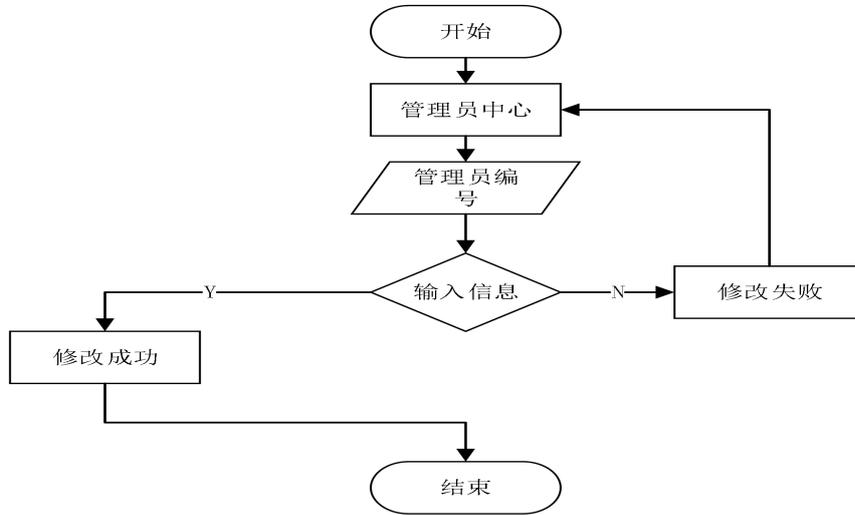


图 4.17 修改信息流程图

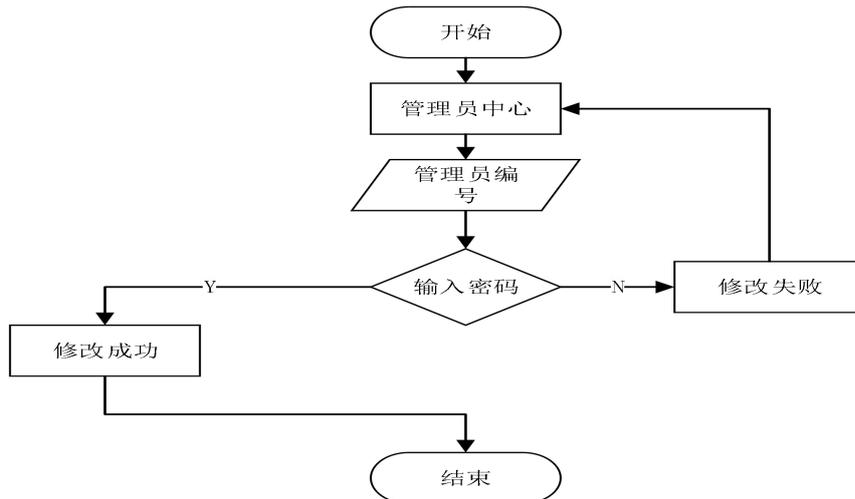


图 4.18 修改密码流程图

点击左侧导航条“管理员中心”->“修改信息”或者“管理员中心”->“修改密码”就可以进入相应的界面。如图 4.19 修改信息图、图 4.20 修改密码图



如图 4.19 修改信息图



图 4.20 修改密码图

修改信息和修改密码相关代码段如下：

```
def MGchangeSelfInfo(request):
    username = request.session.get('username')
    userInfo = User.objects.get(username=username)
    year, mon, day = getIndexData.getNowTime()
    if request.method == 'POST':
        getChangeSelfInfo.changeSelfInfo(username, request.POST, request.FILES)
        userInfo = User.objects.get(username=username)
    return render(request, 'MGchangeSelfInfo.html', {
        'userInfo': userInfo,
        'nowTime': {
```

```

        'year': year, 'mon': getPublicData.monthList[mon - 1], 'day': day,
    }, })
def MGchangePassword(request):
    username = request.session.get('username')
    userInfo = User.objects.get(username=username)
    year, mon, day = getIndexData.getNowTime()
    if request.method == 'POST':
        res = getChangeSelfInfo.getChangePassword(userInfo, request.POST)
        if res != None:
            return ErrorResponse.errorResponse(request, res)
    return render(request, 'MGchangePassword.html', {
        'userInfo': userInfo,
        'nowTime': {
            'year': year,
            'mon': getPublicData.monthList[mon - 1],
            'day': day,
        }, })

```

#### 4.1.8 管理中心

管理中心分为添加景点、景点管理页和用户购票管理页，添加景点则是添加景点的信息，景点管理页则是可以修改地方的景点信息和删除景点信息，用户购票管理页则是可以修改游客的购票信息和删除购票信息。图 4.21 添加景点流程图，图 4.22 景点管理流程图，图 4.23 用户购票管理流程图如下所示：

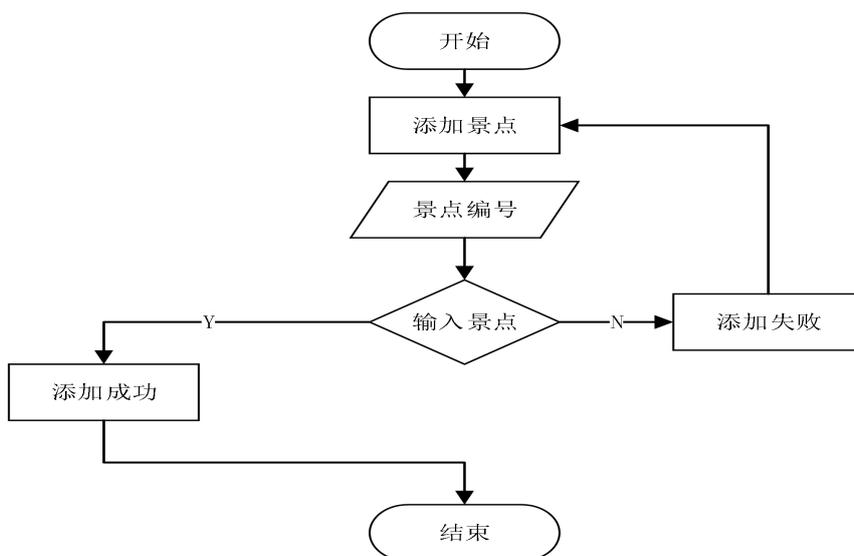


图 4.21 添加景点流程图

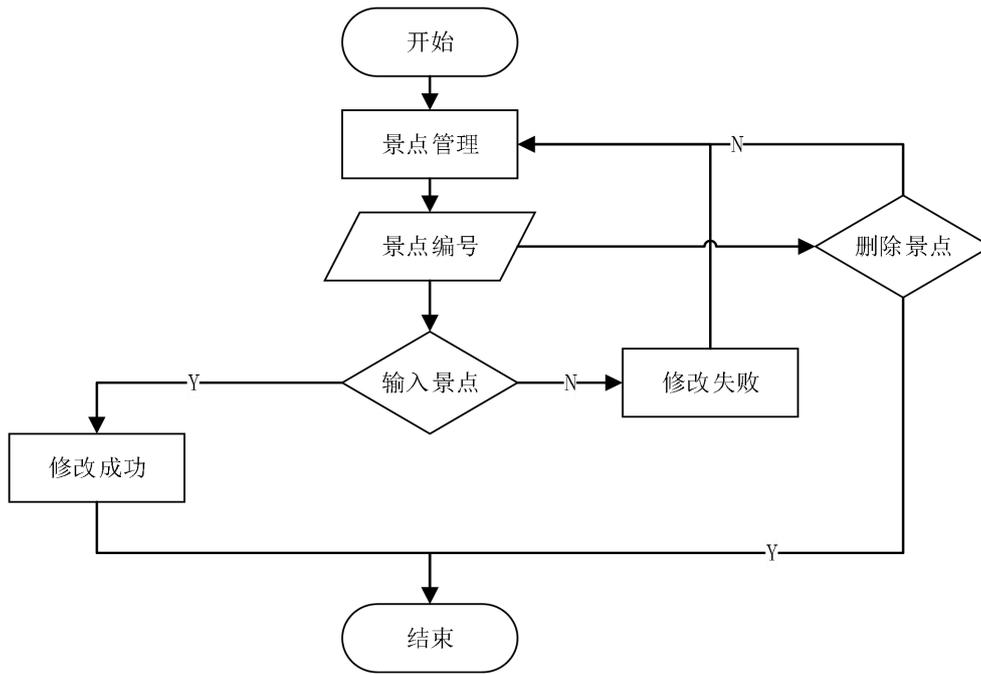


图 4.22 景点管理流程图

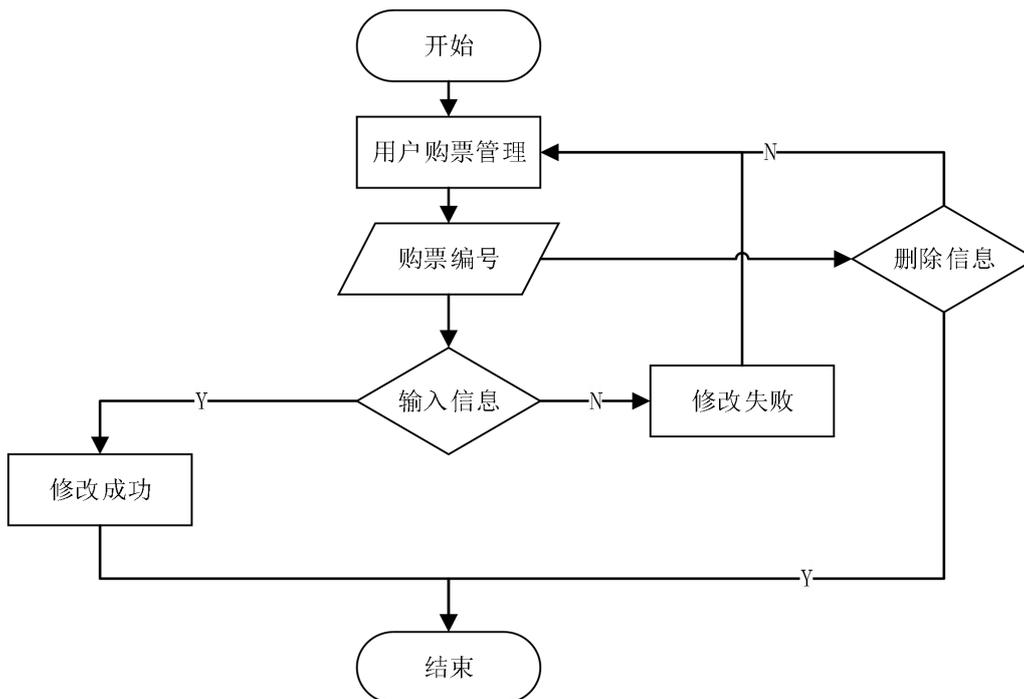


图 4.23 用户购票管理流程图

点击左侧导航条“管理中心”->“添加景点”或->“景点管理页”或->“用户购票管理页”就可以进入相应的界面景点管理页和用户购票管理页都有修改和删除操作。如图 4.24 添加景点图、图 4.25 景点管理图、图 4.26 用户购票管理图、图 4.27 景点管理修改图、图 4.28 用户购票管理修改图所示

首页

管理个人中心

修改信息

修改密码

管理中心

添加景点

景点管理页

用户购票管理页

### 添加景点页面

景区名称:

景点等级:

景区地点:

景区评分:

价格 (元):

销量 (元):

详细地址:

简介:

详细介绍:

[确认](#)

图 4.24 添加景点图

首页

管理个人中心

修改信息

修改密码

管理中心

添加景点

景点管理页

用户购票管理页

### 景点管理页

管理中心 / 景点管理页

2024-June-4

景区名称	景点等级	景区地点	景点评分	价格	销量	操作
凤凰古城	5A景区	凤凰县	5	45 元	888 元	<a href="#">修改</a> <a href="#">删除</a>
芙蓉镇	4A景区	永顺县	5	103 元	910 元	<a href="#">修改</a> <a href="#">删除</a>
城寨奇观旅游区	5A景区	吉首市	5	140 元	24 元	<a href="#">修改</a> <a href="#">删除</a>
凤凰中华大熊猫苑	4A景区	凤凰县	5.0	39 元	72 元	<a href="#">修改</a> <a href="#">删除</a>
乾州古城	4A景区	吉首市	1	49 元	4 元	<a href="#">修改</a> <a href="#">删除</a>
城寨大桥	5A景区	吉首市	5	115 元	40 元	<a href="#">修改</a> <a href="#">删除</a>
德夯大峡谷	5A景区	吉首市	2	40 元	24 元	<a href="#">修改</a> <a href="#">删除</a>
德夯苗寨	4A景区	吉首市	4.7	1 元	59 元	<a href="#">修改</a> <a href="#">删除</a>
竹山苗寨	3A景区	凤凰县	5	98 元	90 元	<a href="#">修改</a> <a href="#">删除</a>
红石林	4A景区	吉丈县	4.7	106 元	62 元	<a href="#">修改</a> <a href="#">删除</a>

图 4.25 景点管理图

首页

管理个人中心

修改信息

修改密码

管理中心

添加景点

景点管理页

用户购票管理页

### 用户购票管理页

管理中心 / 用户购票管理页

2024-June-4

景区名称	预约时间	真实姓名	身份证号	联系方式	操作
凤凰古城	2024-05-31	张三	123456789X	1100	<a href="#">修改</a> <a href="#">删除</a>
古城	2024-06-30	李大炮	123456789X	110	<a href="#">修改</a> <a href="#">删除</a>
曹盘寨	2024-06-02	张老二	43310203012309x	119	<a href="#">修改</a> <a href="#">删除</a>
老5圣地	2024-06-02	张三	43310203012309x	11222	<a href="#">修改</a> <a href="#">删除</a>

图 4.26 用户购票管理图



图 4.27 景点管理修改图

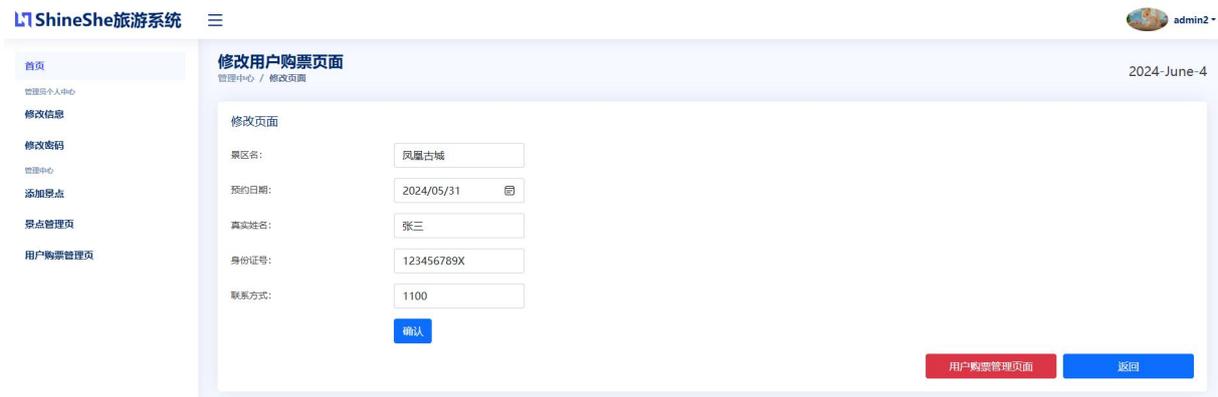


图 4.28 用户购票管理修改图

以下展示添加景点、景点管理、用户购票管理相关的代码段。

添加景点代码段：

```
def MAddJD(request):
    username = request.session.get('username')
    userInfo = User.objects.get(username=username)
    year, mon, day = getIndexData.getNowTime()
    if request.method == 'POST':
        titlefox = request.POST.get('title')
        levelfox = request.POST.get('level')
```

```

provincefox = request.POST.get(' province')
scorefox = request.POST.get(' score')
pricefox = request.POST.get(' price')
saleCountfox = request.POST.get(' saleCount')
detailAddressfox = request.POST.get(' detailAddress')
shortIntrofox = request.POST.get(' shortIntro')
detailIntrofox = request.POST.get(' detailIntro')
if titlefox == '':
    return HttpResponse(' 景区名称不能为空！请返回重新填写！')
elif levelfox == '':
    return HttpResponse(' 景点等级不能为空！请返回重新填写！')
elif provincefox == '':
    return HttpResponse(' 景区地点不能为空！请返回重新填写！')
elif scorefox == '':
    return HttpResponse(' 景区评分不能为空！请返回重新填写！')
elif pricefox == '':
    return HttpResponse(' 价格不能为空！请返回重新填写！')
elif saleCountfox == '':
    return HttpResponse(' 销量不能为空！请返回重新填写！')
elif detailAddressfox == '':
    return HttpResponse(' 详细地址不能为空！请返回重新填写！')
elif shortIntrofox == '':
    return HttpResponse(' 简介不能为空！请返回重新填写！')
elif detailIntrofox == '':
    return HttpResponse(' 详细介绍不能为空！请返回重新填写！')
else:
    Mr = "www.baidu.com"
    TravelInfo.objects.create(
        title=titlefox,
        level=levelfox,
        discount=Mr,
        saleCount=saleCountfox,
        province=provincefox,
        star=Mr,

```

```

        detailAddress=detailAddressfox,
        shortIntro=shortIntrofox,
        detailUrl=Mr,
        score=scorefox,
        price=pricefox,
        commentsLen=Mr,
        detailIntro=detailIntrofox,
        img_list=Mr,
        comments=Mr,
        cover=Mr,
    ) # 创建数据
return render(request, 'MAddJD.html', {
    'userInfo': userInfo,
    'nowTime': {
        'year': year,
        'mon': getPublicData.monthList[mon - 1],
        'day': day,
    },
})

```

景点管理代码段:

```

def MaddressDetail(request):
    username = request.session.get('username')
    userInfo = User.objects.get(username=username)
    MaddressDetailData = TravelInfo.objects.all()
    year, mon, day = getIndexData.getNowTime()
    return render(request, 'MaddressDetail.html', {
        'userInfo': userInfo,
        'MaddressDetailData':MaddressDetailData,
        'nowTime': {
            'year': year,
            'mon': getPublicData.monthList[mon - 1],
            'day': day,
        },
    })

```

景点管理修改景点代码段:

```
def MaddressFox(request):
    username = request.session.get('username')
    userInfo = User.objects.get(username=username)
    year, mon, day = getIndexData.getNowTime()
    nid = request.GET.get('nid')
    MaddressDetailData = TravelInfo.objects.get(id=nid)
    if request.method == 'POST':
        titlefox = request.POST.get('title')
        levelfox = request.POST.get('level')
        provincefox = request.POST.get('province')
        scorefox = request.POST.get('score')
        pricefox = request.POST.get('price')
        saleCountfox = request.POST.get('saleCount')
        detailAddressfox = request.POST.get('detailAddress')
        shortIntrofox = request.POST.get('shortIntro')
        detailIntrofox = request.POST.get('detailIntro')
        if titlefox == '':
            return HttpResponse('景区名称不能为空! 请返回重新填写!')
        elif levelfox == '':
            return HttpResponse('景点等级不能为空! 请返回重新填写!')
        elif provincefox == '':
            return HttpResponse('景区地点不能为空! 请返回重新填写!')
        elif scorefox == '':
            return HttpResponse('景区评分不能为空! 请返回重新填写!')
        elif pricefox == '':
            return HttpResponse('价格不能为空! 请返回重新填写!')
        elif saleCountfox == '':
            return HttpResponse('销量不能为空! 请返回重新填写!')
        elif detailAddressfox == '':
            return HttpResponse('详细地址不能为空! 请返回重新填写!')
        elif shortIntrofox == '':
            return HttpResponse('简介不能为空! 请返回重新填写!')
        elif detailIntrofox == '':
```

```

        return HttpResponseRedirect('详细介绍不能为空! 请返回重新填写!')
    else:
        TravelInfo.objects.filter(id=nid).update(
            title=titlefox,
            level=levelfox,
            saleCount=saleCountfox,
            province=provincefox,
            detailAddress=detailAddressfox,
            shortIntro=shortIntrofox,
            score=scorefox,
            price=pricefox,
            detailIntro=detailIntrofox,
        )
        return redirect("/SS01/MaddressDetail/")
return render(request, 'MaddressFox.html', {
    'userInfo':userInfo,
    'MaddressDetailData':MaddressDetailData,
    'nowTime': {
        'year': year,
        'mon': getPublicData.monthList[mon - 1],
        'day': day,
    },
})

```

景点管理删除景点代码段:

```

def MaddressDetail_delete(request):
    nid = request.GET.get('nid')
    TravelInfo.objects.filter(id=nid).delete()
    return redirect('../')

```

用户购票管理代码段:

```

def userDetails(request):
    username = request.session.get('username')
    userInfo = User.objects.get(username=username)
    userDetailsData = SaleTicketDetail.objects.all()
    year, mon, day = getIndexData.getNowTime()

```

```

return render(request, 'userDetail.html', {
    'userInfo': userInfo,
    'userDetailData': userDetailData,
    'nowTime': {
        'year': year,
        'mon': getPublicData.monthList[mon - 1],
        'day': day,
    },
})

```

用户购票管理修改代码段:

```

def userFox(request):
    username = request.session.get('username')
    userInfo = User.objects.get(username=username)
    year, mon, day = getIndexData.getNowTime()
    nid = request.GET.get('nid')
    SaleTicketDetailw = SaleTicketDetail.objects.get(id=nid)
    if request.method == 'POST':
        titlefox = request.POST.get('titleFox')
        cdayfox = request.POST.get('cdayFox')
        cnameFox = request.POST.get('nameFox')
        cIDcardFoxfox = request.POST.get('IDcardFox')
        connectionfox = request.POST.get('connectionFox')
        if titlefox == '':
            return HttpResponse('景区名不能为空! 请返回重新填写!')
        elif cdayfox == '':
            return HttpResponse('预约时间不能为空! 请返回重新填写!')
        elif cnameFox == '':
            return HttpResponse('真实姓名不能为空! 请返回重新填写!')
        elif cIDcardFoxfox == '':
            return HttpResponse('身份证号不能为空! 请返回重新填写!')
        elif connectionfox == '':
            return HttpResponse('联系方式不能为空! 请返回重新填写!')
    else:
        SaleTicketDetail.objects.filter(id=nid).update(

```

```

        title=titlefox,
        day=cdayfox,

        connection=connectionfox,
        name=cnameFox,
        Idcard=cIDcardFoxfox,
    )

    return redirect("/SS01/userDetail/")
return render(request, 'userFox.html', {
    'userInfo': userInfo,
    'SaleTicketDetailw':SaleTicketDetailw,
    'nowTime': {
        'year': year,
        'mon': getPublicData.monthList[mon - 1],
        'day': day,
    },})

```

用户购票管理删除代码段:

```

def userDetail_delete(request):
    nid = request.GET.get('nid')
    SaleTicketDetail.objects.filter(id=nid).delete()
    return redirect('../')

```

## 4.2 系统性能评价

系统性能分析也是比较重要的内容，进行系统性能分析就是为了确保系统的功能能够在生活中运行使用时，达到规定的指标，因此一个完整的系统软件，是需要进行系统的性能分析这个步骤的。本次进行性能分析主要从易用性指标，可扩展性指标，系统健壮性指标，系统安全性指标这几个方面进行分析。

### 4.2.1 系统的易用性指标

本次程序软件的开发的目的是让使用者可以通过使用该软件提高信息数据的管理效率，同时该程序软件也需要针对不同的操作用户设置对应的功能，因此，此程序的操作流程应该尽量与用户日常操作软件的行为习惯相贴合，另外，程序软件的设计与开发也应考虑非计算机专业用户的计算机操作水平，要让大部分使用者都可以轻松操作程序提供的各个功能。所以此次系统设计的界面非常的简介移动，不需要操作人员进行后台代码的编写，只需要对可视化的界面进行相应的操作，且每个操作都有文字提示。

#### 4.2.2 系统的可扩展性指标

当前需要开发的程序软件是根据当下的用户需求进行设计开发的，但是随着时间的推移，社会大环境的改变，开发出的程序也是需要与时俱进的，需要根据用户不断变换的需求进行相应的功能内容的扩展，需要注意的就是，当对成型的程序进行功能模块新增时，仍然需要保证程序原有架构以及功能不能受到影响，新增的功能模块在系统中也能够运行正常，该指标达标也就可以保证此程序是可以在满足信息管理要求下，从容应对市场环境的变化。所以此次设计使用的软件为 Python、Navicat Premium 15，前者是所有编程语言中最简洁的，在添加相应功能时可以节省非常多的时间，以及代码的冗余。后者是数据库可视化工具，在对 MySQL 进行数据添加时不需要输入大量的代码进行数据的添加，这也减少大量的时间。并且这两者在目前的开发阶段，不会被时代所淘汰。

#### 4.2.3 系统的健壮性指标

程序软件的开发就是为了投入使用时，可以一直稳定的处理各种数据信息，程序软件一旦不稳定，也会给使用者带来干扰。因此从性能分析的角度，就要要求程序软件在应对使用者的误操作，或者是使用者因为各种原因，填写有误的数据等情况时，程序要一直保持稳定，并能够正常让使用者进行使用。当程序的某个功能模块出现异常时，程序的其它功能模块也要能够确保正常使用。程序的健壮性指标达标可以让使用者产生良好的用户体验。所以本次系统设计时，将每个功能模块都单独的设计，当其中一个模块产生错误数据的输入时，并不影响其他模块的使用，提高了系统的健壮性。

#### 4.2.4 系统的安全性指标

程序软件的安全问题是首要问题，毕竟程序对应数据库里面存放的数据信息是庞大的，里面也包括了许多的个人信息，这就对程序要具备一个完善的安全机制提出了要求。因此程序必须要设置登录功能用以进行用户身份的检查，以及身份和权限的匹配，通过对不同用户身份进行功能约束，绝不容忍用户越权操作程序。另外，也需要时刻防范计算机病毒，还有黑客，通过采取针对性的办法进行安全应对，确保程序时刻处于安全的环境，让使用者放心使用。本次系统设计没有用到任何的不良软件，不会在开发过程中植入任何病毒，从源头上解决了此问题。

#### 4.2.5 系统中可能存在的隐患

由于时间有限，本系统没有进行充分的测试，虽然能满足绝大多数的操作需要，但是可能会出现意外情况，还需要维护。例如在输入重复的信息时，给予的是操作成功提示，但数据库会记录该重复信息，导致数据冗余，还需要进一步改进。或者在登录时界面的加载会有一定的延迟。

## 5 总结

通过对 Shine She 旅游系统的开发，让我深刻明白开发一个程序软件需要经历的流程，当确定要开发一个 Shine She 旅游系统的程序时，我在开发期间，对其功能进行合理的需求分析，然后才是程序软件的功能的框架设计，数据库的实体与数据表设计，程序软件的功能详细界面实现，以及程序的功能测试等进行全方位的细致考虑，虽然在此过程中，各个环节都遇到了大大小小的困难，但是通过对这些问题进行反复的分析，深入的思考，借助各种相关文献资料提供的方法与解决思路成功解决面临的各个问题，最后成功的让我开发的 Shine She 旅游系统得以正常运行。

Shine She 旅游系统在功能上面是基本可以满足用户对系统的操作，但是这个程序软件也有许多方面是不足的，因此，在下一个时间阶段，有几点需要改进的地方需要提出来，它们分别是：

(1) 操作页面可以满足用户简易操作的要求，但是在页面多样化设计层面上需要把一些比较丰富的设计结构考虑进来。

(2) 需要对程序的数据结构方面，程序的代码方面等进行优化，让运行起来的程序可以保持稳定运行，也让程序能够保证短时间内处理相关事务，节省处理事务的时间，提高事务处理的效率，同时对服务器上资源占用的比例进行降低。

在开发期间，让我学到了很多的东西，不仅使我在理论上对程序设计有了全新的认识，在实践能力上也得到了提高，真正地做到了学以致用，对我来说受益匪浅。除此之外我还学会了如何更好地和别人沟通，比如不会的问题虚心请教别人。还知道怎么陈述自己的观点。

其次，这次实验我明白了理论知识的重要性。在开始之前，我们需要对相关的理论知识有深入的了解。这些理论知识就像是我们在开发中的指南针，引导我们找到正确的方向。只有深入理解了理论知识，我们才能更好地进行开发，更准确地分析开发结果。在开发过程中也让我看到了编程的严谨性，那种通过实验揭示事物本质的过程，就像是在解开一个个谜团，让我感到无比的兴奋和满足。

最后，这次课设在过程中，总会遇到各种各样的问题，这就需要我们独立思考，寻找解决问题的方法。这个过程虽然充满了挑战，但也让我收获了很多。我学会了如何从不同的角度看待问题，如何利用已有的知识解决新的问题，这些都是我在课设中学到的宝贵的能力。课程设计是一次心灵的触摸与思考，它让我明白了理论知识的重要性，体验到了计算机编程的魅力，学会了独立思考和解决问题的能力，明白了失败的价值。我相信，这次开发的经历将会成为我人生中一笔宝贵的财富。

## 参考文献

- [1] 赵广辉等. Python 程序设计基础[M]. 高等教育出版社, 2021, 5.
- [2] 嵩天等. Python 语言程序设计(第 2 版)[M]. 高等教育出版社, 2017, 2.
- [3] 明日科技, 零基础学 Python[M], 吉林大学出版社, 2018, 4.
- [4] 龚沛曾, 杨志强. Python 程序设计及应用[M]. 高等教育出版社, 2021, 10.
- [5] 叶惠仙, 游金水. Python 语言在大数据处理中的应用[J]. 网络安全技术与应用, 2021, (05): 51-54.
- [6] 汪洋, 姜新通. MVC 框架在 Python 与 Django 下的设计研究[J]. 电脑与信息技术, 2021, 29 (01): 55-57+63.
- [7] 胡钊. 基于 Python Flask 的温度数据可视化[J]. 电子世界, 2021, (08): 117-119.
- [8] 王珊, 张俊. 数据库系统概论(第 5 版)习题解析与实验指导[M]. 高等教育出版社, 2015, 7
- [9] 王雷斌. 计算机机房管理信息系统的分析与设计[J]. 通讯世界, 2020, 27(04): 89-90
- [10] 冉小青. 计算机网络数据库的安全管理技术分析[J]. 计算机产品与流通, 2020(04): 41.