**华东交通大学**

**毕业设计（论文）**

**题目： 基于react的旅游服务平台**

|  |  |  |  |
| --- | --- | --- | --- |
| **学 院:** | **软件学院** | | |
| **专 业:** | **移动开发** | **班 级:** | **2017-3** |
| **学生姓名:** | **黄林** | **学 号:** | **2017211001000920** |
| **指导教师:** | **苏琦** | **完成日期:** | **2021年2月26日** |

|  |
| --- |
| **毕业设计（论文）诚信声明**  **本人郑重声明：所呈交的毕业设计（论文）是我个人在导师指导下进行的研究工作及取得的研究成果。就我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表和撰写的研究成果，也不包含为获得华东交通大学或其他教育机构的学位或证书所使用过的材料。**  **如在文中涉及抄袭或剽窃行为，本人愿承担由此而造成的一切后果及责任。**  本人签名  导师签名  **2021 年 2 月 26日** |

**华东交通大学毕业设计（论文）任务书**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 黄林 | | | 学号 | | 2017211001000920 | | | | 毕业届别 | 21 | | 专业 | | 软件工程（移动开发） |
| 毕业设计（论文）题目 | | | | | 基于react的旅游服务平台 | | | | | | | | | | |
| 指导教师 | | 苏琦 | | | | | 学历 | | 本科 | | | 职称 | | 工程师 | |
| **具体要求：**  1、设计目的与内容  包含选题意义、设计目的（如系统描述）、设计内容（如系统功能描述等）  设计目的：现在网络时代，信息是很重要的，很多人出去旅游，会有很多因素的影响，比如交通问题，赶不上发车时间导致只能买新的车票，因为旅游景点门票卖完了，或者价格的原因，有些人可能觉得不划算或者预算超支。路由系统，可以实现路程的查询，车票的详细信息查询，可以在很久之前就安排好出门的各种路线和时间规划，还有旅游的信息查询，旅游点的详细游玩，也可以看看别人的评价和分享。自己安排好游玩的计划，能合理的安排时间，也能更加详细的了解周边游玩的资料，不用担心到了地点后一片迷茫，只能跟着大众走，没有自己的安排。旅游点负责人，可以将自己的旅游点多介绍出去，也能为自己多揽些客人，当别人买票前来游玩，也能给别人多介绍介绍景点和路线。当想开发自己的更多景点，也能将自己的景点和介绍挂在系统上，可以将新加景点放在平台热门处，吸引更多的顾客。平台管理员，可以和商家合作，将商家的付费内容挂在用户最容易看到的地方，吸引顾客。可以去中国各地去探查，给一些好的地方代言，介绍出去，给那些地方带来收益，也能改善当地经济收入。  设计内容：  1:登录注册模块  用户注册自己的账号，设置自己的密码，保存数据，完成注册，之后可以进行登录，使用自己注册的账号密码，验证正确登录成功，登录成功，进入主页  2:客服模块  用户点击主页的联系客服，会进入服务中心界面，主要有首页问题查询，帮助中心，联系客服，投诉，反馈建议等动能模块，首页问题查询，会细分问题类型，有会员服务，酒店相关，机票相关，旅游度假，门票相关等问题查询。投诉模块，会给你显示投诉方式和途径。反馈建议会给你选择反馈方向和原因，登录自己的账号，方便回复。联系客服是展示旅游平台的客服电话  3:机票模块  可以选择机票的起点和终点，搜索选定时间的所有机票，可以购买单程或者往返的机票，每一张机票还会有详细介绍，当你购买后，会显示在你的订单中，展示你的出发时间，到达时间  4:酒店模块  酒店老板发布自己的酒店住房信息和价格，以及房间的照片和配置，游客搜索自己想去的地方，查询当地的酒店，通过货比三家，查看以前顾客的评价和照片，选择合适的酒店预定，到时间后，直接去酒店办理入住，酒店根据入住用户的身份证信息判断入住人员预定的房间类型  5:度假模块  2、设计要求与成果  包含设计工具、原始数据、技术要求（定性和定量要求，如性能指标）、设计步骤、英文翻译、设计成果（毕业论文、图表如电路图、源码、实物样品等）  包含设计工具、原始数据、技术要求（定性和定量要求，如性能指标）、设计步骤、英文翻译、设计成果（毕业论文、图表如电路图、源码、实物样品等）  设计工具：vscode和mongodb可视化工具  技术要求：react和node加mongodb数据库开发  设计步骤：需求分析，数据库设计，页面使用react确定和设计，后端接口编写  英文翻译：找一篇与毕设选题相关的英文研究论文，要求2000以上，将其翻译成中文  设计成果:设计页面主要是采用react和antd实现的单叶应用，主页面是一个tab页，可以进入不同的模块，首页上方有登录注册的链接，还有查看订单和联系客服，左侧是一个个人信息的固定弹窗，可以隐藏和张开，内部有登录信息的，订单，账户，消息等。tab页有每一个模块，每一个模块进去可以进行相应的操作。按时完成系统的设计和毕业论文的设计  3、应收集的主要参考文献  [1]阿兰，德波顿 机场里的小旅行。  [2]吴伟民。数据结构。 清华大学出版社  [3]戴特尔 C How to Program,Sixth Edition. 电子工业出版社  [4]王德新。计算机组成。 复旦大学出版社  [5]刘金玲。 数据库原理与应用。 清华大学出版社  [6]佛兰纳根。 The Definitive Guide ,Sixth Edition 机械工业出版社  **进度安排：**   |  |  |  |  | | --- | --- | --- | --- | |  | 设计各阶段内容 | 时间 | % | | 1 | 选题和资料收集阶段 | 第7学期16～17周 | 5 | | 2 | 分析计划阶段 | 第7学期19～20周 | 10 | | 3 | 设计阶段 | 第8学期1～3周 | 25 | | 4 | 实现和测试阶段 | 第8学期4～7周 | 40 | | 5 | 毕业论文写作、查重、答辩 | 第8学期8～12周 | 20 | | 6 | 英文资料翻译 | 第7学期19～20周 |  |   指导教师签字：  2021年2月28日 | | | | | | | | | | | | | | | |
| 题目发出日期 | | | 2021.1.19 | | | | | 设计（论文）起止时间 | | | 2021.1.19-2021.4.14 | | | | |
| 学院意见：  同意发布题目  毕业设计领导小组组长签章 | | | | | | | | | | | | | | | |

**华东交通大学毕业设计（论文）开题报告书**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课题名称 | 基于react的旅游服务平台 | | | | |
| 课题来源 |  | 课题类型 |  | 导师 |  |
| 学生姓名 |  | 学号 |  | 专业 |  |
| **一、开题报告内容**  1、目的和意义  为了给想出去旅游的人一个更好的信息介绍，想出去玩，最怕的是行程问题和对旅游景点的迷惑，这个系统的实现可以实现对更多用户去了解旅游景点更大的帮助，也可以帮助一些穷苦地方，想开发自己的家乡，增加收益，把自己家乡介绍出去  意义：（补充完整）  帮助大家更舒心的旅游，帮助旅游的人的舒心和顺利，帮助穷苦地方的旅游开发，防止有些旅游景点蒙骗用户，从旅游角度讲，游客可以通过该系统，便捷，及时的了解到不同景点的信息，景点周边的酒店信息，而且可以实现门票和酒店的预订，从管理人员角度讲，管理者可以通过该系统及时地添加景点，酒店的信息，，可以对游客的酒店订单进行管理，还可以查看游客的评论，该课题的研究实现不仅方便了游客，而且大大地提高了管理人员的管理效率  2、研究现状和发展趋势  研究现状：（补充完整）  就目前旅游企业的现状来看,旅游企业普遍属于规模小、作坊式的中小企业  其信息化建设尚处于初期发展阶段:还有相当一部分旅游企业是运用传统的商业  模式,伯息服务能力非常差,完全不能满足旅游者的需求。开发旅游资讯系统已  经成了旅游业发展不可逆转的趋势。旅游管理信息系统的便捷性、低成本、覆盖  面广等优势是传统旅游管理方式不可比拟的。  目前,旅游信息管理网络化程度较低,有的没有、甚至无法连接互联网。通  过对信息资源的深度开发和广泛利用,不断提高内部信息管理、资源合作、决策  效率和水平,从而更好地为游客提供服务。如今,旅游景点信息的网络化已经成  为全球各大知名景点公认的现代化管理模式，因而，本人设计饼开发网络化的旅游管理系统  发展趋势：（补充完整）  网络营销的实施是必然的，我国网络营销一直处于高峰时期，人才，认识等各方面的成长，传统营销渠道以及策略已经不能满足国人的消费，主导地外慢慢变成网络营销，清华大学总裁班网络营销专家，知名讲师刘东明表示，对于传统行业，一步跨越显然不现实，正确做法是吧网络营销和传统营销结合，旅游系统，改变了传统营销的限制，可以随时随地了解  3、设计方案  3.1功能设计  系统用户分为两类角色:系统管理员和会员。会员主要实现6个功能:会员  注朋、景点查询、门票订购、酒店查询、消店预订、景点评论。系统管理员主要  实现:会员管理、景点类型管理、景点管理、省份管理、门票审核、门票统计  酒店管理、预订审核、预订统计、评价管理,管理员的功能结构  3.2 数据库的概念设计  概念结构设计的任务是将数据库需求分析得到的用户数据需求抽象为信息  结构,是整个数据库设计的关键。概念结构设计的口标是产生反映信息需求的整  体数据库概念结构,即概念模式  根据对系统的分析,系统的实体有管理员实体、景点实体、门票实体、酒店  实体、房间实体、会员实体等。  根据项目需要完成的功能、项目的具体需求以及关系数据库的实体完整性,  数据表可以设计为:  用户表 sys user(id, user name, user\_pwd, full name, sex, cardno, teladdr, type)  省份表area(d,name)  景点类型表 spot type(id,mame)  景点信息表spot(id,mame, typed, areal, contentI, content2, price  门票订购表 ticket(id,code, spoted, userId, re gate, amount, price, total state)  酒店信息表 hotel( id, name, yepename, spoted, content  房间类型表 roomtype(id, hotell  name, price  房间信息表 room(ic, roomno, typed,ftor)  房间预订表book(id,code, roomed, userid, regrate,days, outdate, price, total,  state, remark)  景点评论表 comment(id, spoted, userId, sendtime, content)  3.3数据库的逻辑结构设计」  数据库的逻辑结构由ER图转换成相对应的关系模式。根据关系型数据库规  范化理论对关系模式进行相应的规范化处理,达到BCNF范式。该系统共设计了  10张数据表,通过数据库中各个数据表之间的关联,为系统的前台功能和后台管  理功能提供支持。 | | | | | |

课题类型：（1）A—工程设计；B—技术开发；C—软件工程；D—理论研究；

（2）X—真实课题；Y—模拟课题；Z—虚拟课题

（1）、（2）均要填，如AY、BX等。此部分可以附页

**华东交通大学毕业设计（论文）开题报告书（续）**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4、进度安排   |  |  |  |  | | --- | --- | --- | --- | |  | 设计各阶段内容 | 时间 | % | | 1 | 选题和资料收集阶段 | 第7学期16～17周 | 5 | | 2 | 分析计划阶段 | 第7学期19～20周 | 10 | | 3 | 设计阶段 | 第8学期1～3周 | 25 | | 4 | 实现和测试阶段 | 第8学期4～7周 | 40 | | 5 | 毕业论文写作、查重、答辩 | 第8学期8～12周 | 20 | | 6 | 英文资料翻译 | 第7学期19～20周 |  |   5、主要参考文献  1]阿兰，德波顿 机场里的小旅行。  [2]吴伟民。数据结构。 清华大学出版社  [3]戴特尔 C How to Program,Sixth Edition. 电子工业出版社  [4]王德新。计算机组成。 复旦大学出版社  [5]刘金玲。 数据库原理与应用。 清华大学出版社  [6]佛兰纳根。 The Definitive Guide ,Sixth Edition 机械工业出版社   1. **方法及预期目的**   1、研究方法（手段）  查看各旅游平台的设计风格和内容，作为自己的参考，设计页面需求，再自己设计构思一些页面，结合需求来设计页面。在以往的多次出去游玩风景区，自己的心得体会，加上各平台的游客评价，还有旅游场地的现场调研询问游客的评价。研究和分析出旅游平台的需求和设计  2、预期目的  设计一个旅游服务平台，能提供一位游客旅游时候的每一种需求，保证游客在这个平台操作就能实现旅游的安排。设计一个个好看的和方便操作的界面  **三、指导老师意见**  同意开题  指导教师签名： 日期：2021.2.28 |

**华东交通大学毕业设计(论文)评阅书(1)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** | 黄林 | **学号** | 2017211001000920 | **专业** | 软件工程（移动开发） |
| **毕业设计(论文)题目** | | **基于react的旅游服务平台** | | | |
| **指导教师评语：**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 具 体 要 求 | 优 | 良 | 中 | 一般 | 差 | | 出勤及工作态度（20%） |  |  |  |  |  | | 方法合理，设计工作量饱满（20%） |  |  |  |  |  | | 论点正确，论文内容有一定难度（20%） |  |  |  |  |  | | 结构严谨，论文有一定应用价值（20%） |  |  |  |  |  | | 对前人工作有改进或有独特见解。（10%） |  |  |  |  |  | | 论文格式正确，撰写规范（10%） |  |  |  |  |  |  |  |  | | --- | --- | | **得分** |  |   **指导教师签字：**  **2021年 月 日**  **评阅人评语：**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 具 体 要 求 | 优 | 良 | 中 | 一般 | 差 | | 选题合理，设计有应用价值（20%） |  |  |  |  |  | | 方法合理，设计工作量饱满（20%） |  |  |  |  |  | | 论点正确，论文内容有一定难度（20%） |  |  |  |  |  | | 对前人工作有改进或有独特见解。（20%） |  |  |  |  |  | | 论文格式正确，撰写规范（20%） |  |  |  |  |  |  |  |  | | --- | --- | | **得分** |  |   **评阅人签字：**  **2021年 月 日** | | | | | |

**华东交通大学毕业设计(论文)评阅书(2)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** | 黄林 | **学号** | 2017211001000920 | **专业** | 软件工程（移动开发） |
| **毕业设计(论文)题目** | | **基于react的旅游服务平台** | | | |
| **答辩小组评语：**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 具 体 要 求 | 优 | 良 | 中 | 一般 | 差 | | 符合要求 |  |  |  |  |  | | 答辩准备充分，论文题目与内容相符 |  |  |  |  |  | | 语言精练能突出重点，思路清晰能准确表达 |  |  |  |  |  | | 论点正确，论文内容有一定难度 |  |  |  |  |  | | 方法合理，论文内容工作量饱满 |  |  |  |  |  | | 结构严谨，论文有一定应用价值 |  |  |  |  |  | | 对前人工作有改进或有独特见解 |  |  |  |  |  | | 正面回答问题 |  |  |  |  |  | | 回答问题有理论依据，基本概念清楚 |  |  |  |  |  | | 主要问题回答准确，深入 |  |  |  |  |  |  |  |  | | --- | --- | | **得分** |  |   **组长签字：**  **2021年 月 日** | | | | | |
| **答辩委员会意见：**  **同意以上评定，根据前面三项得分，按25:15:60的比例评定总成绩为**   |  |  | | --- | --- | | **等级** |  |   **进入二次答辩 □ 评优组 最终成绩由二次答辩决定**  **□ 合格组**  **答辩委员会主任签字：**  **2021年 月 日（学院公章）** | | | | | |

注：答辩小组根据评阅人的评阅签署意见、初步评定成绩，交答辩委员会审定，盖学院公章。

“等级”用优、良、中、及、不及五级制（可按学院制定的毕业设计(论文)成绩评定办法评定最后成绩）。

**华东交通大学毕业设计（论文）答辩记录**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 黄林 | 学号 | 2017211001000920 | 毕业届别 | 2021 | 专业 | | 软件工程（移动开发） |
| 题目 | 基于react的旅游服务平台 | | | | 答辩时间 | | 2021年 月 日 | |
| 答辩组成员（签字）： | | | | | | | | |
| 答辩记录：  记录人（签字）：    答辩小组组长（签字）： | | | | | | | | |

基于react的旅游服务平台的设计和实现

摘 要

随着社会经济的不断发展，越来越多的人兴趣开好广泛，喜欢到全国乃至世界各地走走，看看历史文化和风景名胜。但是出门旅游的规划一直是每一个想旅游的人心中的难点。路线规划，旅游行程，旅游时的吃住行都是问题，不及时规划好，不预定好房间和吃饭的地方。旅游景点一直是人多热闹场地，可能就会导致没有合适的吃住地方

本文首先分析林旅游服务平台以及旅游意义的背景，明确林本文的研究思路和方向。然后按照软件工程项目的要求，进行林企业的业务需求分析，并通过业务流程整理和问题分析，整理出系统的需求，根据分析结果进行林系统的建模，在需求分析的基础上，进行林系统的设计，包括软件应用架构设计，业务逻辑设计，其间，以面向对象的理论为基础，运用UML进行系统的相关设计，明确了在B/S模式下采用二层架构来实现目标系统，在系统的分析和设计的基础上，进行系统的数据库设计，包括数据库建模（E-R图）和数据库逻辑设计，最后，对系统进行来开发和运行测试，通过react，node，axios等技术的运用，开发来一个用户交互性能良好，功能实用，能满足当前需求以及可扩展的旅游服务平台系统，并对开发的系统进行了相关的测试

在本文的研究中，在系统的架构设计方面，以及用户交互体验方面进行了较多的工作，为了实现一个扩展性强，代码重用度高，以便将来的扩展和维护，

本文主要开发的功能有：登录注册，机票模块，门票模块，景点模块，度假模块，火车票模块

关键词：旅游服务平台系统；二层架构；axios

**Design and implementation of tourism service platform based on react**

（上一行写自己的中文题目对应的英文题目）

**Abstract**

With the continuous development of social economy, more and more people have a wide range of interests, like to go around the country and even around the world to see the history, culture and scenic spots. But the planning of travel has always been a difficult point for everyone who wants to travel. Route planning, travel itinerary, food, accommodation and travel are all problems. If we don't plan in time, we don't reserve rooms and places to eat. Tourist attractions have always been crowded and busy places, which may lead to no suitable places to eat and live

This paper first analyzes the background of forest tourism service platform and tourism significance, and makes clear the research ideas and direction of forest tourism. Then, according to the requirements of the software engineering project, the business requirements of forest enterprises are analyzed, and the system requirements are sorted out through business process arrangement and problem analysis. According to the analysis results, the forest system is modeled. On the basis of the requirements analysis, the forest system is designed, including software application architecture design and business logic design. In the meantime, based on the object-oriented theory Based on the analysis and design of the system, the database design of the system is carried out, including database modeling (E-R diagram) and database logic design. Finally, the system is developed and tested. Through the use of react, node, Axios and other technologies, the system is developed We have sent a tourism service platform system with good user interaction, practical functions, which can meet the current needs and can be expanded, and have carried out relevant tests on the developed system

In this paper, we have done a lot of work in the system architecture design and user interaction experience, in order to achieve a strong scalability, high code reuse, for future expansion and maintenance,

The main functions of this paper are: login registration, ticket module, ticket module, scenic spot module, vacation module, train ticket module

Key words: tourism service platform system; two tier architecture; Axios目 录

**（目录采用word的自动生成目录方式，不能自己添加页码。）**

[1 绪 论 1](#_Toc318664910)

[1.1 研究的背景及意义 1](#_Toc318664911)

[1.1.1 选题的背景 1](#_Toc318664912)

[1.1.2 国内外研究现状 1](#_Toc318664913)

[1.1.3 研究的意义 2](#_Toc318664914)

[1.2 系统目标 2](#_Toc318664915)

[2 需求分析 3](#_Toc318664916)

[2.1 功能需求 3](#_Toc318664917)

[2.1.1 子系统/模块说明 3](#_Toc318664918)

[2.1.2 功能需求描述 3](#_Toc318664919)

[2.2 非功能需求 3](#_Toc318664920)

[2.2.1 外部接口需求 3](#_Toc318664921)

[2.2.2 性能需求 3](#_Toc318664922)

[2.2.3 其它需求 3](#_Toc318664923)

[3 总体设计 5](#_Toc318664924)

[3.1 运行环境 5](#_Toc318664925)

[3.2 基本处理流程 5](#_Toc318664926)

[3.3 模块结构 5](#_Toc318664927)

[3.4 外部接口 5](#_Toc318664928)

[3.5 内部接口 5](#_Toc318664929)

[4 数据库设计 6](#_Toc318664930)

[4.1 概念结构设计 6](#_Toc318664931)

[4.1.1 设计思路 6](#_Toc318664932)

[4.1.2 E-R图 6](#_Toc318664933)

[4.2 逻辑结构设计 6](#_Toc318664934)

[4.2.1 设计思路 6](#_Toc318664935)

[4.2.2 逻辑模型 6](#_Toc318664936)

[4.3 物理结构设计 6](#_Toc318664937)

[4.3.1 存取方式 6](#_Toc318664938)

[4.3.2 存储结构 6](#_Toc318664939)

[5 界面设计 7](#_Toc318664940)

[5.1 界面关系图或工作流图 7](#_Toc318664941)

[5.2 界面设计成果 7](#_Toc318664942)

[5.2.1 主界面 7](#_Toc318664943)

[5.2.2 子界面 7](#_Toc318664944)

[6 详细设计 8](#_Toc318664945)

[6.1 系统主要功能模块介绍 8](#_Toc318664946)

[6.2 模块一设计 8](#_Toc318664947)

[6.2.1 模块一算法描述 8](#_Toc318664948)

[6. 2.2 模块一程序流程图 8](#_Toc318664949)

[6. 2.3 模块一关键类说明 8](#_Toc318664950)

[6.3 模块二设计 8](#_Toc318664951)

[6.3.1 模块二算法描述 8](#_Toc318664952)

[6.3.2 模块二程序流程图 8](#_Toc318664953)

[6.3.3 模块二关键类说明 8](#_Toc318664954)

[…… 8](#_Toc318664955)

[7 编码 9](#_Toc318664956)

[7.1 代码实现与核心算法 9](#_Toc318664957)

[7.2 代码优化分析 9](#_Toc318664958)

[8 测试 10](#_Toc318664959)

[8.1 测试方案设计 10](#_Toc318664960)

[8.1.1 测试策略 10](#_Toc318664961)

[8.1.2 测试进度安排 10](#_Toc318664962)

[8.1.3 测试资源 10](#_Toc318664963)

[8.1.4 关键测试点 10](#_Toc318664964)

[8.2 测试用例构建 10](#_Toc318664965)

[8.2.1 测试用例编写约定 10](#_Toc318664966)

[8.2.2 测试用例设计 10](#_Toc318664967)

[8.2.3 关键测试用例 10](#_Toc318664968)

[8.2.4 测试用例维护 10](#_Toc318664969)

[9 总结与展望 11](#_Toc318664970)

[9.1 设计工作总结 11](#_Toc318664971)

[9.2 未来工作展望 11](#_Toc318664972)

[谢 辞 12](#_Toc318664973)

[参考文献 13](#_Toc318664974)

[附录A 外文翻译—原文部分 15](#_Toc318664975)

[附录B 外文翻译—译文部分 16](#_Toc318664976)

[附录C 软件使用说明书 17](#_Toc318664977)

[附录D 主要源代码 18](#_Toc318664978)

[附录E 相关图表 19](#_Toc318664979)

1 绪 论

（以上为一级标题，黑体三号加粗，居中）

(……..此处为论文内容)

## 1.1 研究的背景及意义

（以上为二级标题，黑体四号加粗，居左顶格）

(…….. 此处为论文内容)

### 1.1.1 选题的背景

（以上为三级标题，黑体小四，居左顶格）

现在网络时代，信息是很重要的，很多人出去旅游，会有很多因素的影响，比如交通问题，赶不上发车时间导致只能买新的车票，因为旅游景点门票卖完了，或者价格的原因，有些人可能觉得不划算或者预算超支。路由系统，可以实现路程的查询，车票的详细信息查询，可以在很久之前就安排好出门的各种路线和时间规划，还有旅游的信息查询，旅游点的详细游玩，也可以看看别人的评价和分享。自己安排好游玩的计划，能合理的安排时间，也能更加详细的了解周边游玩的资料，不用担心到了地点后一片迷茫，只能跟着大众走，没有自己的安排。旅游点负责人，可以将自己的旅游点多介绍出去，也能为自己多揽些客人，当别人买票前来游玩，也能给别人多介绍介绍景点和路线。当想开发自己的更多景点，也能将自己的景点和介绍挂在系统上，可以将新加景点放在平台热门处，吸引更多的顾客。平台管理员，可以和商家合作，将商家的付费内容挂在用户最容易看到的地方，吸引顾客。可以去中国各地去探查，给一些好的地方代言，介绍出去，给那些地方带来收益，也能改善当地经济收入。

……（以上为范文示例）

### 1.1.2 国内外研究现状

(…….. 此处为论文内容)

旅游业是中国对外开放最早的行业，也是发展最快的行业之一，旅游业已经成为世界上发展最快和规模最大的产业之一，在全世界范围内，受社会，经济，和科学发展的共同推进，旅游业的发展步伐还在继续加快，我国旅游业作为改革开放的急先锋得到突飞猛进的发展，但是在发展的时还存在一些问题，

### 1.1.3 研究的意义

帮助大家更舒心的旅游，帮助旅游的人的舒心和顺利，帮助穷苦地方的旅游开发，防止有些旅游景点蒙骗用户，从旅游角度讲，游客可以通过该系统，便捷，及时的了解到不同景点的信息，景点周边的酒店信息，而且可以实现门票和酒店的预订，从管理人员角度讲，管理者可以通过该系统及时地添加景点，酒店的信息，，可以对游客的酒店订单进行管理，还可以查看游客的评论，该课题的研究实现不仅方便了游客，而且大大地提高了管理人员的管理效率

## 1.2 系统目标

设计一个旅游服务平台，能提供一位游客旅游时候的每一种需求，保证游客在这个平台操作就能实现旅游的安排。设计一个个好看的和方便操作的界面

2 需求分析

## 2.1 功能需求

### 2.1.1 子系统/模块说明

1:登录注册模块，2：机票模块，3:酒店模块，4：火车票模块，5:度假模块，6:门票模块，7:当地人模块

8:攻略模块，9:订单模块，10:客服模块，11:后台管理模块

### 2.1.2 功能需求描述

1:用户模块

主要实现了用户注册、用户登录、用户注销、用户个人信息更新管理以及主页展示等功能。

2:机票模块

进行查询起点到终点的每一个时间段的机票，当登录之后，可以进行购买，更换，退票功能

3:酒店模块

查询整晚的房间和钟点房，提供晚上睡觉和一段时间的休息。用户可以根据居住地进行搜索所有适合的房间和时间。还可以挑选不同的房间类型，比如大床房和情侣房

4:火车票模块

可以查看出发到终点的火车票，登录之后可以买火车票，且购买的票乘坐时间不能有冲突

5:度假模快

可以直接搜索景点，搜索地点，景点名，景点类型。也可以直接选自己的地点，推荐一些景点数据，会推荐景点的购票，套餐票

6:门票模块

根据主题，目的地，景点搜索每个景区的门票的详细信息，门票价格和不同类型的门票

7:当地人模块

会根据你搜索的旅游景点，推荐你合适的包车游览，接送机，当地顾问等。还可以搜索当地玩法

8:订单模块是根据你在本系统购买的所有吃喝玩住行的票的记录

9:客服模块可以让你在平台及时反馈和沟通询问

10:后台管理模块是给景点商家的一个更新景点信息，添加景点游玩项目的后台，可以设置不同的介绍内容，改变门票价格等等关于游玩的介绍

## 2.2 非功能需求

### 2.2.1 外部接口需求

本系统在本地机上进行开发、测试。需要主机及内存容量：内存256M及以上；外存容量、联机或脱机、媒体及其存储格式，设备的型号及数量：硬盘40G及以上、声卡、显卡无要求；

硬件接口需求：

表 2-2-1 硬件接口需求表

|  |  |  |  |
| --- | --- | --- | --- |
| 硬件接口名称 | 硬件名称 | 厂商 | 接口描述 |
| 主机 | MAC笔记本 | apple | 用于开发、运行软件 |

软件接口需求：

表 2-2-2 软件接口需求表

|  |  |  |  |
| --- | --- | --- | --- |
| 软件接口名称 | 外部组件名称 | 版本号 | 接口描述 |
| 数据库 | Mongodb | 8.0 | 用于数据库建立 |
| 操作系统 | 任意 | 任意 | node可运行任何操作系统 |
| 工具软件 | VScode |  | 用于前端后台开发 |

通信接口需求：

表 2-2-3 通信接口需求表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 通信接口名称 | 协议或方式 | 安全要求 | 传输速率要求 | 同步通信描述 |
| Restful接口 | HTTP | 秘密 | 512KB/S | / |

### 2.2.2 性能需求

该系统的主要的性能主要体现在mongodb数据库增删改查的速度，以及前端页面第一次加载和解析的时间

用户界面：采用div+css的通用图形界面，对用户友好，且必须对鼠标和键盘提供支持，界面的设计应如下：

要具有一致性，界面规范应遵循web软件界面的规范

提供简单的错误处理

提供信息反馈，用多种信息提示用户当前软件运行状态，软件界面的功能

显示画面，画面简洁明快，富有现代气息，不能太过花哨

各管理功能的不同模块在不同的界面，

数据精确度：

查询时候应保证查全率，查准率，

时间特性

一般操作时间应保持响应时间在3～5秒内，以及数据的导入导出也应该在相应的时间内

### 2.2.3 其它需求

可维护性：

1. 系统如果出现服务器关闭，数据库连接断开等软件异常，会给在log信息详细标示；
2. 如果事务处理过程出现故障，事务回滚，保证数据的完整性；
3. 如果多人同时访问一条记录，加锁该纪录避免脏数据。
4. 数据库采用增量备份法，出现数据库故障时数据库恢复。

可移植性：

本系统可在Windows及linux的操作系统中运行。

易用：界面应该为用户提供方便，软件的界面应该符合目前留下的界面规范

可维护：项目出现问题能很快探测哪里出了问题，及时改正

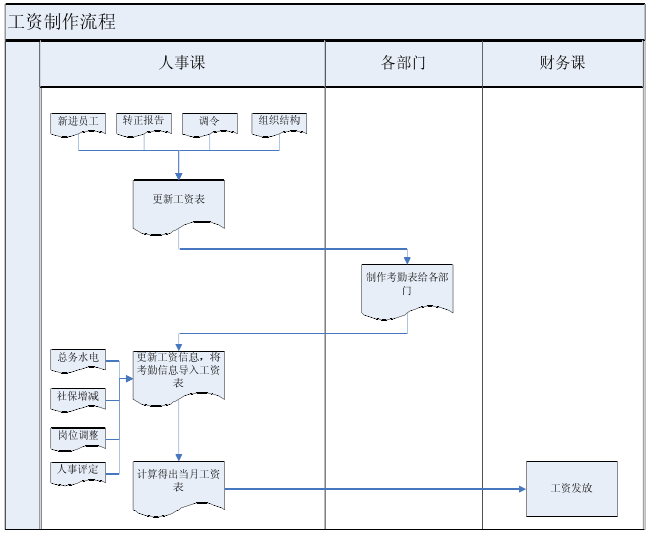


图 2-1 工资制作流程图

（论文中***图*** 的画法，注意以下几点：1 图要用word画，或者采用粘贴方式，不能使用截图方式，否则打印出来图的效果很不清晰。2 ***图的标题*** 要写在***图的下方***，***居中***，用***5号黑体字***。3 ***图中的文字***采用***5号宋体字***。4 ***每章***的所有图采用***顺序编号***，如图2-1，图2-2等）

表 2-2 工资管理问题点

|  |  |  |  |
| --- | --- | --- | --- |
| 涉及点 | 目前做法 | 存在问题 | 希望改善 |
| 考勤 | 各部门汇总 | 考勤存在虚报、错误时无法核对，导致错误。 | 考勤准确 |
| 工资计算 | 凭公式计算 | 灵活，但易出错，且不易检查。 | 固定某些必要内容，使出错率降低。 |
| 人员信息 | 以批注的形式予以提醒 | 易出现遗漏现象，且人事系统与工资系统无法相对应。 | 使用同一系统，达到信息共享目的。 |

（论文中***表*** 的画法，注意以下几点：1 表要用word画，不能使用截图方式，否则打印出来效果很不清晰。2 ***表的标题*** 要写在***表的上方***，***居中***，用***5号黑体字***。3 ***表中的文字*** 采用***5号宋体字***。4 ***每章*** 的所有表采用 ***顺序编号***，如表2-1，表2-2等）

3 总体设计

## 3.1 运行环境

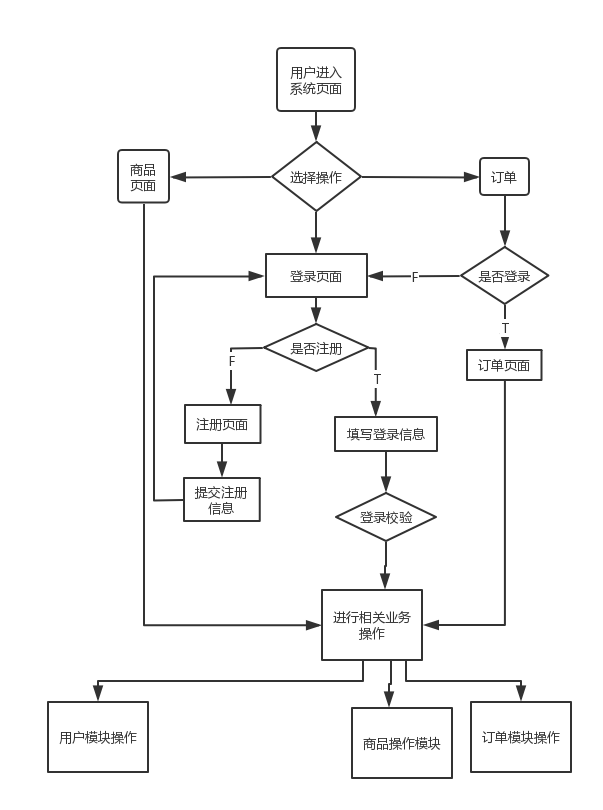
（说明对本系统的运行环境,包括硬件环境和支持环境的规定。）

硬件环境：笔记本电脑

软件环境：vscode软件，node12.17.0 npm6.14.4 谷歌浏览器 mongodb数据库

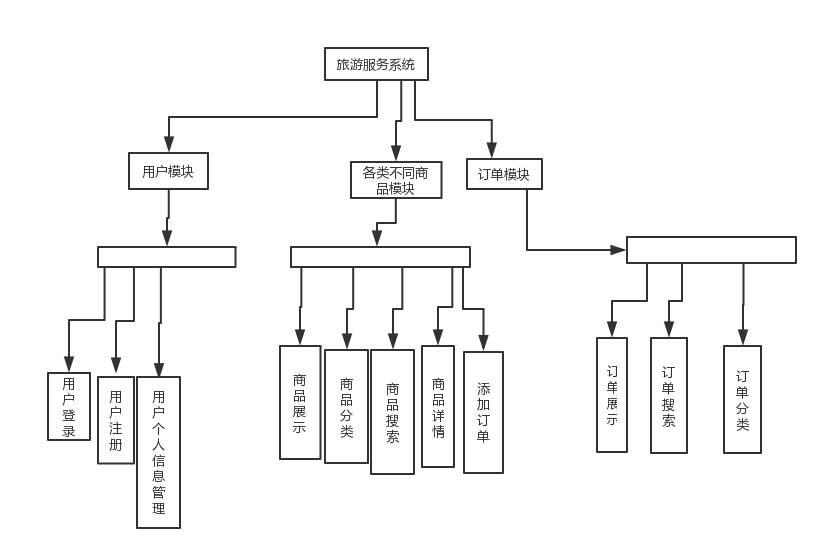
## 3.2 基本处理流程

（说明本系统的基本处理流程，尽量使用图表的形式。）



## 3.3 模块结构

（用一览表及框图的形式说明本系统的系统元素（各层模块、子程序、公用程序等）的划分，扼要说明每个系统元素的[标识符](http://baike.baidu.com/view/390932.htm" \t "_blank)和功能，分层次地给出各元素之间的控制与被控制关系。）



## 3.4 外部接口

（说明本系统同外界的所有接口的安排包括软件与硬件之间的接口、本系统与各支持软件之间的接口关系。）

本系统暂时未使用外部接口

## 3.5 内部接口

（说明本系统之内的各个系统元素之间的接口的安排。）

4 数据库设计

## 4.1 概念结构设计

### 4.1.1 设计思路

1根据用户需求来设计数据库，需求收集和分析是设计的第一步，

2根据资料分析进行概念结构设计，概念结构设计是整个数据库设计的关键，概念结构独立于数据库逻辑结构，能充分反应现实世界，包括实体和实体之间的联系，能满足用户对数据处理的要求

3据库物理设计

对一个给定的逻辑数据模型选取一个最适合应用环境的物理结构的过程，称为数据库的物理设计，所谓数据库的物理结构主要指数据库在物理设备上的存储结构和存储方法，

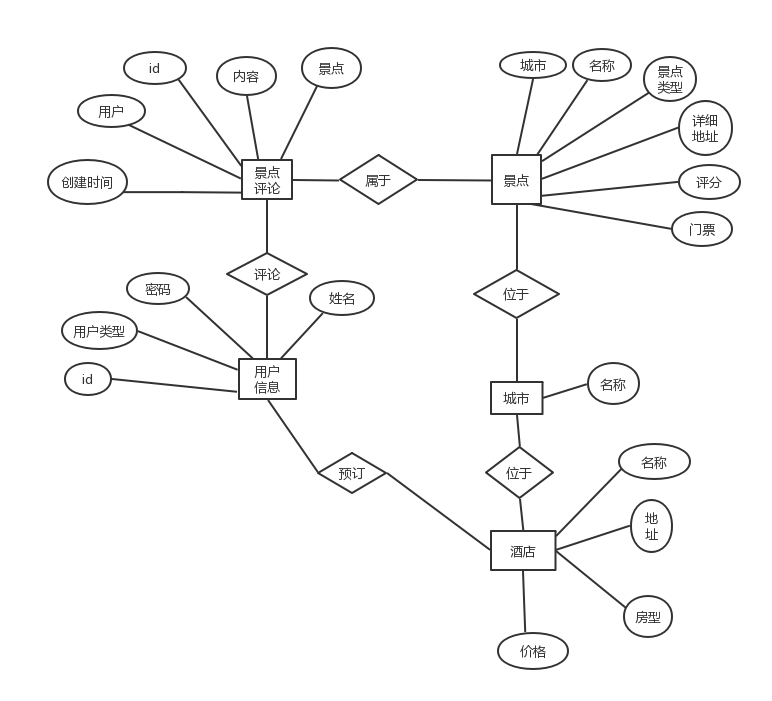
4数据库实施

对数据库的物理设计初步评价完成后就可以建立数据库，数据库的实施响应于软件工程的编码，调试阶段，

5数据库运行和维护

数据库投入运行标志着开发任务的基本完成和维护的开始，但并不意味着设计完成，任何数据库他存在一天，他的设计就得不断的进行评价，调整，修改，甚至于完全改变

### 4.1.2 E-R图



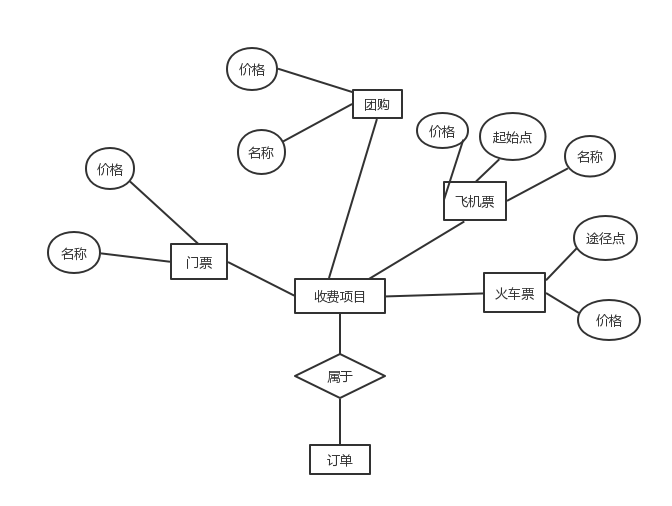


图1-1E-R关系图

## 4.2 逻辑结构设计

### 4.2.1 设计思路

通过对本系统的基本功能需求分析，可以看出本系统的结构相对来说比较简洁明了，

按照我所设定好的移动营业厅应用系统的所有功能要求，主要设计了以下四个数据库表：

表4.2.1.1用户信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 |
| Id | Float | 4 | 编号 |
| Uswrname | String | 10 | 用户名 |
| Password | String | 15 | 密码 |
| Type | Int | 1 | 用户类型 |

表4.2.1.2飞机票数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 |
| Id | Float | 4 | 编号 |
| startCity | String | 4 | 出发地 |
| startTime | Date | 10 | 出发时间 |
| arriverCity | String | 4 | 到达地 |
| arriverTime | Date | 4 | 到达时间 |
| Money | Int | 4 | 价格 |

### 表4.2.1.3酒店商家信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 |
| Id | Float | 4 | 编号 |
| Name | String | 4 | 商家姓名 |
| Location | String | 8 | 酒店地址 |
| Number | Float | 2 | 酒店评分 |
| Data | Object | 1 | 房间信息 |

表4.2.1.4火车票数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 |
| Id | Float | 4 | 编号 |
| name | String | 6 | 列车名 |
| startTime | Date | 8 | 起始时间 |
| arriverTime | Date | 8 | 到达时间 |
| money | Array | 3 | 三种位置价格 |
| RemainingNum | Array | 3 | 三种位置剩余数量 |
| CrossCity | Array | 20 | 途径城市 |
| CrossTime | Array | 20 | 途径城市时间 |

表4.2.1.5景点信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 |
| Id | Float | 4 | 编号 |
| Name | string | 20 | 景点名称 |
| startCity | String | 10 | 出发地点 |
| StartTime | Date | 8 | 出发时间 |
| Type | Number | 1 | 景点类型 |
| Location | String | 20 | 景点地址 |
| Money | Number | 10 | 价格 |
| Score | Number | 2 | 评分 |
| Data | Object | 1 | 详细景点介绍 |

表4.2.1.7度假

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 |
| Id | Float | 4 | 编号 |
| Name | String | 20 | 项目名 |
| Money | Long | 4 | 价格 |
| Score | shotInt | 2 | 评分 |
| Location | String | 10 | 地址 |
| Type | array | 4 | 类型 |

表4.2.1.9团购

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 |
| Id | Float | 4 | 编号 |
| Name | String | 20 | 项目名 |
| Money | Long | 4 | 价格 |
| Score | shotInt | 2 | 评分 |
| Detail | Int | 2 | 详细介绍索引 |

表4.2.1.10团购详情

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 |
| childId | Int | 2 | 索引id |
| crossCity | Array | 10 | 经过城市 |
| crossName | Array | 10 | 经过项目名 |
| crossDetail | Array | 10 | 经过项目介绍 |

### 4.2.2 逻辑模型

（1 E-R图转换为关系表，2 逻辑模型优化，3 用户视图设计）

关系表：

User(id, name, password, type)

planeTicket(id, startCity, startTime, arriverCity, arriverTime, money)

Hotel(id, name, location, number, data)

trainTicket(id, name, startTime, money, arriverTime, RemainingNum, CrossCity, CrossTime)

scenicSpot(id, name, startCity, startTime, type, location, money, score, data)

Vacation(id, name, money, score, location, type)

groupBuy(id, Name, money, score, detail)

groupChild(childId, crossCity, crossName, crossDetail)

## 4.3 物理结构设计

### 4.3.1 存取方式

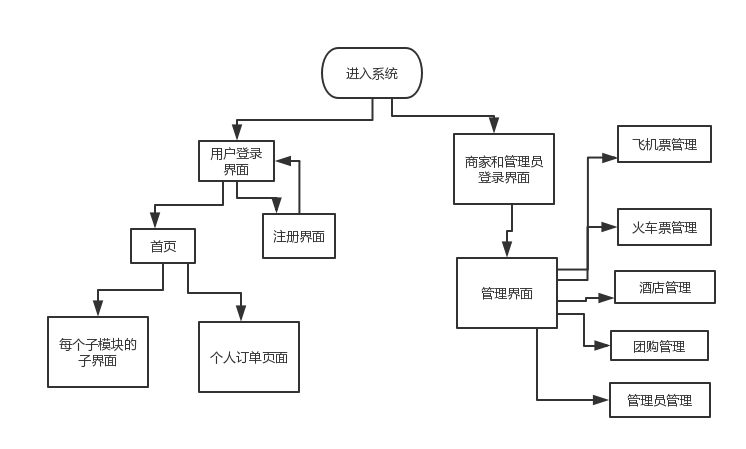
我设计的表结构里面使用了索引，当我们查询普遍的时候，那么就可以设置索引加快查询速度，并且可以避免全表查询

### 4.3.2 存储结构

我设计的l旅游服务系统的数据库采用的是顺序存储结构，虽然顺序存储修改麻烦，但是查询效率高

5 界面设计

## 5.1 界面关系图或工作流图



## 5.2 界面设计成果

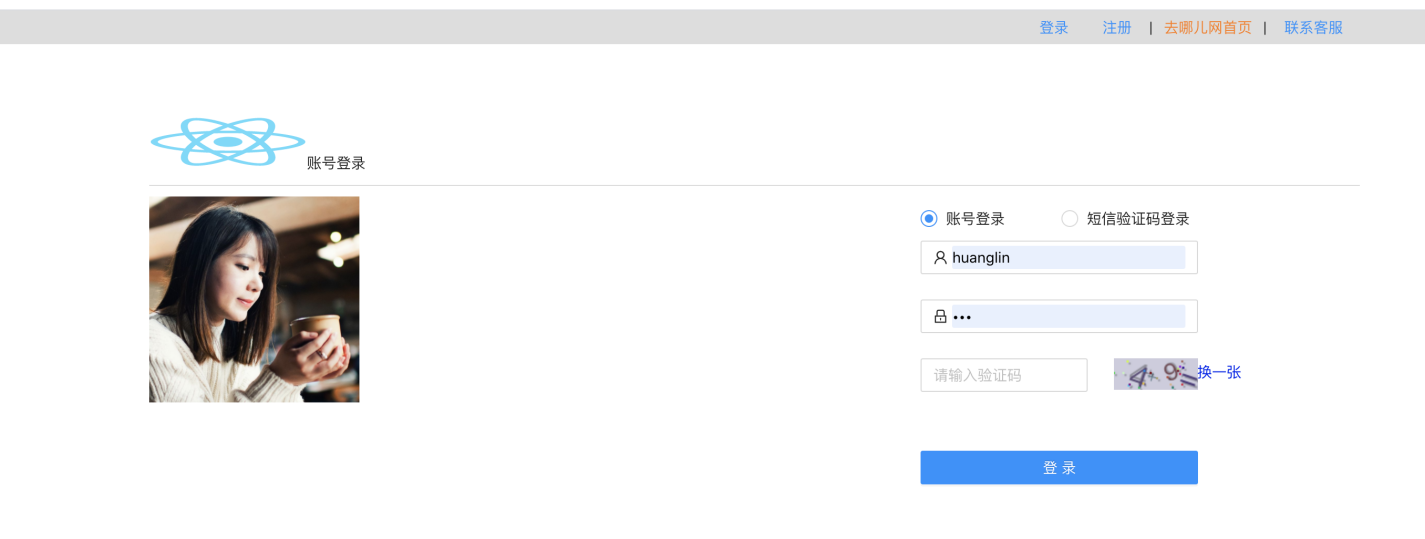
### 5.2.1 主界面

主页面



### 5.2.2 子界面

### 5.2.2.1登录界面



5.2.2.2机票界面



5.2.2.3火车票



5.2.2.4酒店



5.2.2.4度假



**订单**



注册界面



门票



当地人

# **6 详细设计**

## 6.1 系统主要功能模块介绍

首先我们设计的是

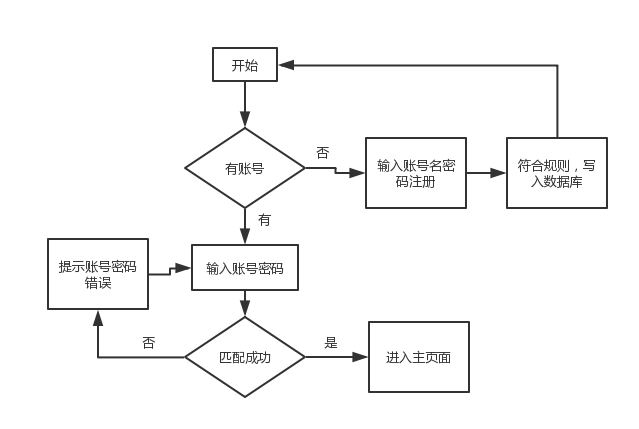
1. 用户模块，这就包括的是普通用户的登录注册，管理员登录，
2. 订单模块：所有旅游相关的购买记录
3. 火车票模块：火车详细信息和购买退票功能
4. 酒店模块：酒店的信息介绍，地点分类
5. 机票模块：机票单程的信息，购票退票功能
6. 度假模块：度假的相关套票，分类查询，套票信息和购买
7. 门票模块：景点查询，地点分类，不同价位购买
8. 当地人模块：主要是当地游玩和WIFI租赁服务

## 6.2 模块一登录注册设计

### 6.2.1 模块一算法描述

在现实使用中，往往进入界面都需要账号密码匹配数据库数据才能进入，如果是第一次进入页面，就需要创建一条账号数据，也就是注册。注册为了保密，需要账号名和自己设计的规定格式的密码，以便在登录时候使用，通过用户输入用户名，密码，确认密码，调用接口，/login传入参数，注册时候，存入数据库，登录时候，再与数据库中信息进行匹配，匹配失败则登录失败，匹配成功，则进入主页面

### 6.2.2 模块一程序流程图



### 6.2.3 模块一关键类说明

function (err, db) {

if (err) throw err;

var dbo = db.db("admin");

var whereStr = data; // 查询条件

//find是查询条件，limit是返回条数

dbo

.collection("user")

.find(whereStr)

.limit(1)

.toArray(function (err, result) {

if (err) throw err;

db.close();

if (result[0]) {

let rule = {

id: result[0].\_id,

telephone: result[0].telephone,

password: result[0].password,

};

jwt.sign(rule, "Bearer", { expiresIn: 3600 }, function (

err,

token

) {

if (err) throw err;

res.json({

status: 0,

token: token,

name: whereStr.telephone,

});

});

} else {

res.json({

status: 1,

message: "账号名或密码错误",

});

}

});

}

Admin admin创建用户实体类，通过/login传入参数

Var dbo = db.db(“admin”);

whereStr是传入参数

调用方法dbo.collection("user").find(whereStr)得到列表

判断数据与数据库是否匹配，如果不匹配返回错误，匹配就返回成功，

注册时通过/register传入参数，查询用户信息表，

Var dbo = db.db(“admin”);

whereStr是传入参数

调用方法dbo.collection("user").find(whereStr)得到列表

判断是否有相同用户名，不存在，则存入数据库中，返回注册成功提示

## 6.3 模块二机票模块设计

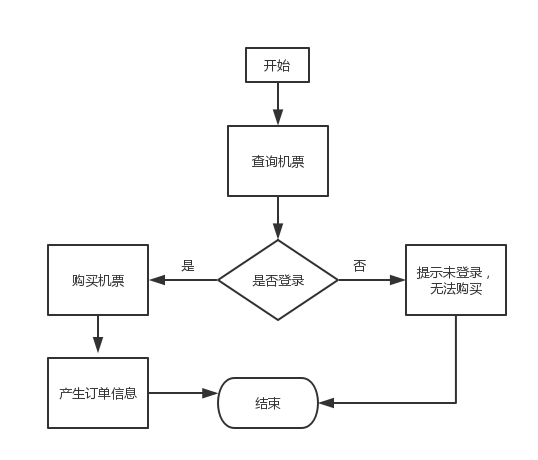
### 6.3.1 模块二算法描述

查询：通过对出发地和到达地的筛选，查询机票，还可以通过选择是否往返，把往返的合适车票查询出来，

修改：当机票有位不可抗力原理产生的延迟，可以由机票管理人员进行人工更改航班的时间，调用/planeUpdate/{name}对特定航班进行修改

删除：由于天气问题，或者其他特殊原因导致航班不能启程，可由管理人员删除该航班，调用/planeDelete/{name}对特定航班进行删除

### 6.3.2 模块二程序流程图



### 6.3.3 模块二关键类说明

创建用户实体类，

查询：通过/planeTicket传入参数

Var dbo = db.db(“admin”);

whereStr是传入参数

调用方法dbo.collection("plane").find(whereStr)得到列表

删除：通过/planeDelete/{name}指定飞机票进行删除

购买：通过/planeMoney/{id}/{name}通过特定的飞机票id和用户名name，购买飞机票，

Var dbo = db.db(‘admin’);

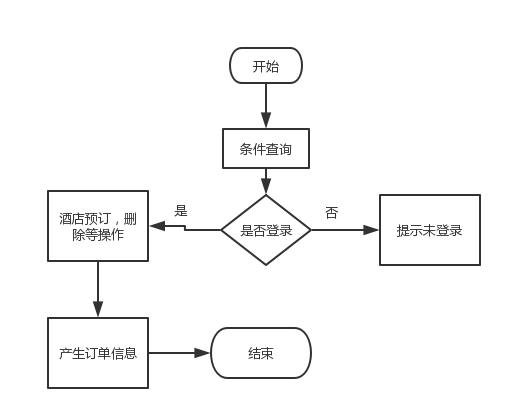
调用方法dbo.collection(“plane”).find(whereStr)购买飞机票，删除数据库plane指定数据，再把飞机票信息加到指定用户name的订单中，购买成功，返回订单信息，失败，提示问题

## 6.4 模块三酒店模块设计

### 6.4.1 模块三算法描述

现实生活中，需要用户登录，通过地址，价格区间，酒店类型，评级等查询酒店，查询出指定条件区间的酒店，然后通过预订酒店指定房间，付款后就可以在指定时间去入住，预订之后，会在订单中出现预订信息，也可以退订

### 6.4.2 模块三程序流程图



### 6.4.3 模块三关键类说明

unction (err, db) {

if (err) throw err;

var dbo = db.db("admin");

var whereStr = data; // 查询条件

//find是查询条件，limit是返回条数

dbo

.collection("hotel")

.find(whereStr)

.toArray(function (err, result) {

if (err) throw err;

db.close();

res.send(result);

});

}

通过一些条件，查询hotel表的数据，得到符合条件的酒店信息列表，

查询：/hotelSearch/{string},

Var dbo = db.db(admin)

Dbo.collection(“hotel”).find(where)

查询得到列表

预订：想预订酒店，通过/hotelMoney/{string}预订酒店，再同步给订单添加信息，接口/addToData/{hotelD}

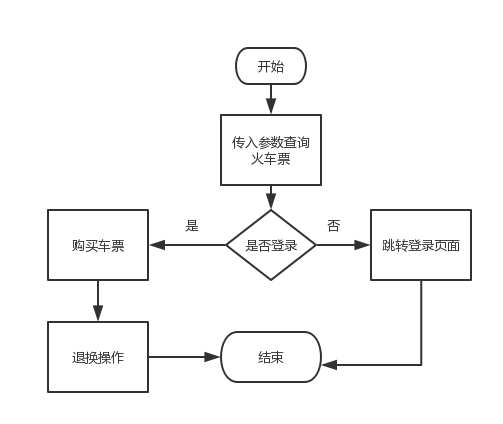
删除：删除酒店订单，使用/hotelDetele/{string}删除酒店指定订单信息

## 6.5 模块四火车票模块设计

### 6.5.1 模块四算法描述

现实中火车票的购买，退票，修改车票时间等，只需要在规定时间前就可以操作，通过起始点查询火车票，选购所需时间的火车票，然后购买，可以在规定时间前换，也可以退票，通过接口/trainSearch查询，通过/trainMoney购买，通过trainDelete退订，传入指定的火车名和日期就可以

### 6.5.2 模块四程序流程图



### 6.5.3 模块四关键类说明

function (err, db) {

if (err) throw err;

var dbo = db.db("admin");

var whereStr = data;

dbo

.collection("trainTicket")

.find(whereStr)

.toArray(function (err, result) {

if (err) throw err;

db.close();

res.send(result);

});

}

火车票有查询购买，换票，退票操作

查询：通过/trainSearch/{time}通过时间查询火车票，

var dbo = db.db(admin)

Dbo.collection(“train”).find(where)

购买：通过/trainMoney，是在登录之后才可以进行的操作传入用户名和车次id。

var dbo = db.db(admin)

Dbo.collection(“train”).find(where)购买车次，再使用addToData将车票加入订单

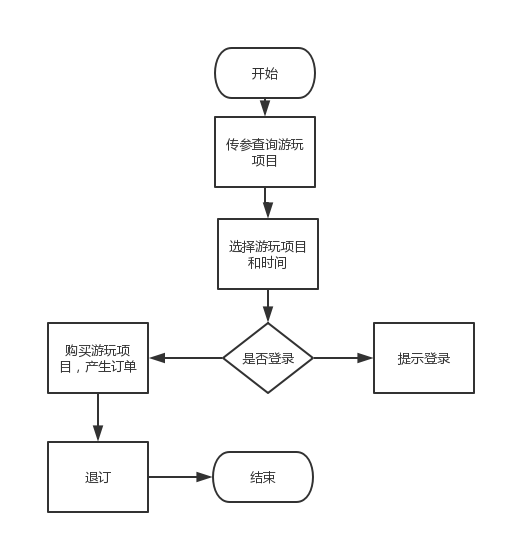
退改票：通过/trainDelete传入车次信息，删除指定订单，再使用购票接口，就是改车票了

## 6.6 模块五度假设计

### 6.6.1 模块二算法描述

现实生活中有很多城市的不同游玩项目，通过选一个城市，筛选出游玩项目，度假，就是购买一个游玩全场票，会有规划路线，和价格等，通过/HolidaySearch传入城市查询度假的项目，选择一个项目，可以查询项目信息，选择指定时间就可以购买，，通过/HolidayMoney购买项目后，会在订单信息中产生记录，可以使用/HolidayDetele传入订单信息，进行退订。

### 6.6.2 模块二程序流程图



### 6.6.3 模块二关键类说明

使用/trainSearch/{city}查询度假项目的全部项目，

Var dbo = db.db(‘admin’);

Dbo.collection(“train”).find(where)查询train表的所有数据列表

选择一个项目后，选定时间，进行/trainMoney/{time}/{id}传入项目id和游玩时间time，

Var dbo = db.db(‘admin’);

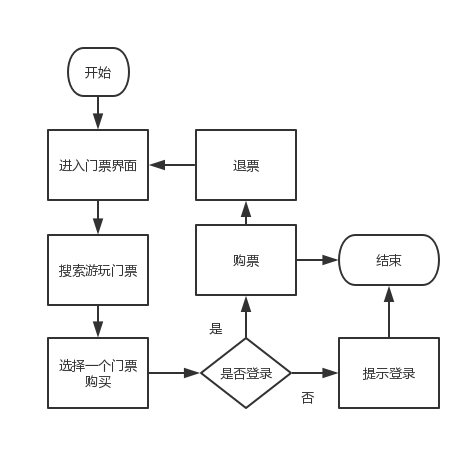
Dbo.collection(‘user’)存入项目记录和时间

## 6.7 模块六门票模块设计

### 6.7.1 模块二算法描述

现实生活中，门票是一个游玩地的准许证明，购买门票，有游玩和价格等介绍，门票购买了，是在指定时间可以进景点游玩，可以查询不同城市，不同景点的不同游玩票。通过/ticketSearch查询门票，然后通过/ticketMoney购买门票，通过/ticketDelete退票

### 6.7.2 模块二程序流程图



### 6.7.3 模块二关键类说明

function (err, db) {

if (err) throw err;

var dbo = db.db("admin");

var whereStr = data; // 查询条件

//find是查询条件，limit是返回条数

dbo

.collection("tickets")

.find(whereStr)

.toArray(function (err, result) {

if (err) throw err;

db.close();

res.send(result);

});

}

通过调用查询接口/ticketSearch/{string}/{pageSize}查询一个游玩地点的门票

Var dbo = db.db(admin)连接admin库，dbo.collection(‘ticket’).find(where)查询ticket表数据,传入pageSize表示数据分页的，返回指定数量和第几条开始返回

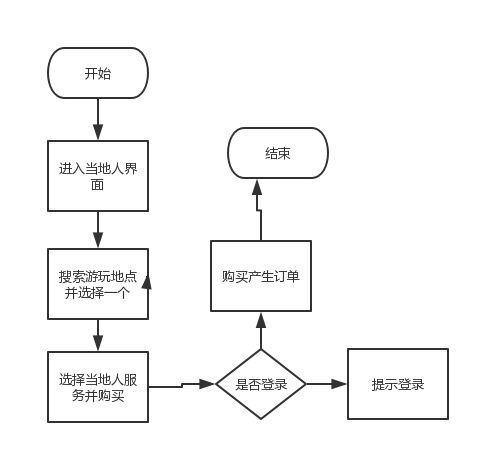
选择一个门票进行购买，使用/TicketMoney{id}进行购买，并将数据保存在订单信息中

## 6.8 模块七当地人模块设计

### 6.8.1 模块二算法描述

当地人，简单意思就是旅游地原住人，对本地最熟悉的人，当你想好目的地后，可以在本地租赁一切需要的东西，使用/localSearch/{string}查询一个景点，可以进行wifi租赁，/wifiMoney，租车服务，/vehicelMoney进行车辆服务，带你走遍各地，可以选择/themeSearch搜索主题游，由当地人带领玩乐

### 6.8.2 模块二程序流程图



### 6.8.3 模块二关键类说明

function (err, db) {

if (err) throw err;

var dbo = db.db("admin");

var whereStr = data; // 查询条件

//find是查询条件，limit是返回条数

dbo

.collection("local")

.find(whereStr)

.toArray(function (err, result) {

if (err) throw err;

let data = result;

db.close();

if (whereStr.sort && whereStr === "top") {

result = sort(data);

}

if (whereStr.sort && whereStr === "top") {

res.send(result);

} else {

result.reverse();

res.send(result);

}

});

}

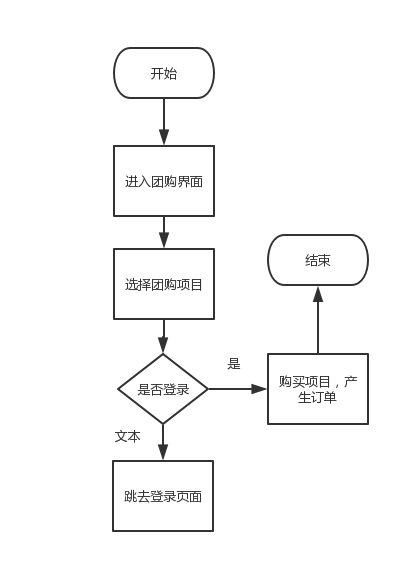
使用/localSearch/{str}当地人项目，使用var dbo = db.db(“admin”)连接库，查询local表Dbo.collection(‘local),然后使用/localMoney购买当地人项目，

## 6.9 模块八团购模块设计

### 6.9.1 模块二算法描述

旅游的团购项目，选择几天几夜的，跟着旅游团，一切被安排好的游玩，使用/groupBuy/{str}进行搜索团购项目，使用/groupBuyMoney购买团购

### 6.9.2 模块二程序流程图



### 6.9.3 模块二关键类说明

unction (err, db) {

if (err) throw err;

var dbo = db.db("admin");

var whereStr = data; // 查询条件

//find是查询条件，limit是返回条数

dbo

.collection("groupBuying")

.find(whereStr)

.toArray(function (err, result) {

if (err) throw err;

db.close();

res.send(result);

使用/groupBuy/{str}查询团购项目，使用var dbo = db.db(“admin”)连接库，查询/group表Dbo.collection(‘group’),然后使用/groupBuyMoney购买团购项目

7 编码

## 7.1 代码实现与核心算法

冒泡排序

function bubbleSort(arr) {

var len = arr.length;

for (var i = 0; i < len; i++) {

for (var j = 0; j < len - 1 - i; j++) {

if (arr[j] > arr[j + 1]) {

//相邻元素两两对比

var temp = arr[j + 1]; //元素交换

arr[j + 1] = arr[j];

arr[j] = temp;

}

}

}

return arr;

}

对数据进行排序，将数据库查出来的数据排序返回

jwt.sign(rule, "Bearer", { expiresIn: 3600 }, function (

err,

token

) {

if (err) throw err;

res.json({

status: 0,

token: token,

name: whereStr.telephone,

});

});

node查询数据，使用jwt模块返回数据

vue算法

function patch (oldVnode, vnode) {

// some code

if (sameVnode(oldVnode, vnode)) {

patchVnode(oldVnode, vnode)

} else {

const oEl = oldVnode.el // 当前oldVnode对应的真实元素节点

let parentEle = api.parentNode(oEl) // 父元素

createEle(vnode) // 根据Vnode生成新元素

if (parentEle !== null) {

api.insertBefore(parentEle, vnode.el, api.nextSibling(oEl)) // 将新元素添加进父元素

api.removeChild(parentEle, oldVnode.el) // 移除以前的旧元素节点

oldVnode = null

}

}

// some code

return vnode

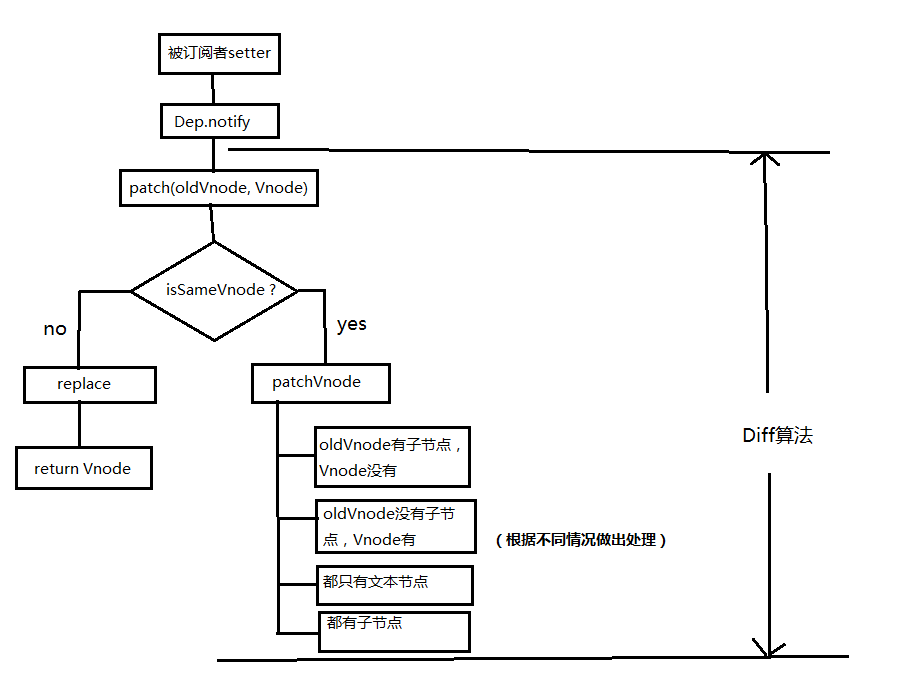
}

patch函数接收两个参数oldVnode和Vnode分别代表新的节点和之前的旧节点

判断两节点是否值得比较，值得比较就执行patchVnode

不知道比较就用Vnode替换oldNode

如果两个节点是一样的，那就比较他们的子节点，如果两个节点不一样，那就证明被改变了，直接替换



## 7.2 代码优化分析

function selectionSort(arr) {

let len = arr.length;

let minIndex, temp;

for(let i = 0; i < len - 1; i++) {

minIndex = i;

for(var j = i + 1; j < len; j++) {

if (arr[j] < arr[minIndex]) {

minIndex = j;

}

}

temp = arr[i];

arr[i] = arr[minIndex];

arr[minIndex] = temp;

}

return arr;

}

console.time()

selectionSort(array)

console.timeEnd()

使用代码进行测试算法的效率，优化算法的空间复杂度和时间复杂度，正确性，可读性，健壮性都是算法的评定标准，

算法优化：

随机搜索，利用随机数求极小点而求得函数近似的最优解的方法

梯度下降法：是一个最优化的算法，通常称为最速下降法，

登录时候的密码加密功能，

图片的懒加载和路由的懒加载

项目的构建打包

8 测试

## 8.1 测试方案设计

### 8.1.1 测试策略

后台代码：

后台代码一般是使用vscode软件编写的，运行的时候，node的框架exprot结合vscode的插件，会自动把问题提示出来在控制台，使用chai断言库，使用内部的assert模块，判断数据类型等

前端代码：

使用vue的单元测试，也可以使用vscode自带的插件进行测试，在代码中使用debugger，配合谷歌浏览器的测试，进行断点测试代码，查看代码的数据流程

### 8.1.2 测试进度安排

代码实现完成，安排测试，进行页面样式检查，尽量保持风格一致，检查接口调试，数据返回与数据库是否一致

### 8.1.3 测试资源

数据库数据，自己去网上找材料，进行制作

### 8.1.4 关键测试点

测试

## 8.2 测试用例构建

### 8.2.1 测试用例编写约定

使用mochajs.org作为测试框架，配合vscode使用，按照mocha模块

### 8.2.2 测试用例设计

引入assert，

Describe(‘Array’, function(){

Describe(#indexof(), function() {

It(‘should return -1 when the value is not present’, function() {

[1,2,3].indexof(5).should.equal(-1)

[1,2,3].indexof(0).should.equal(-1)

})

})

})

### 8.2.3 关键测试用例

### 8.2.4 测试用例维护

9 总结与展望

## 9.1 设计工作总结

在设计基于react的旅游服务平台，通过设计各个功能，登录模块，飞机票模块，度假模块，团购模块等，我学到了很多平时学不到的东西，特别是团购模块，需要很多的搜索条件，需要很多的表关联起来，更加考验了我对系统架构的理解，了解create-react框架，减少程序员开发难度，还有在编码方面学会了如何更加简洁明了的编写代码，一些以前没有注意的编码规范，例如驼峰显示，字母大小写等，总的来说，这次毕设让我学会了很多，将专业知识转化为实践，在以后的工作中，我会充分利用这些经验，做一名更好的程序员

## 9.2 未来工作展望

目前虽然应用基本功能都以实现，但是还是略显粗糙，有些功能模块设计的不严

谨。

①登录模块，后续加入验证码登录模块，增加系统安全和严谨

②系统用户管理模块：有一些固定的字段写死，不可以随意修改。

③业务信息管理模块：后续快要到结束时间没有完成的业务，增加短信告警功能，

达到强提醒的效果。

④界面优化：界面设计的不够简洁，后期要继续美化。

⑤密码修改模块，后期加入加密解密功能，增加系统的安全性。

⑥业务管理模块：界面设计的还是很粗糙，有些字段设置的不够全面具体。后期

要加入更严谨的考虑，例如增加完成任务按钮，增加任务具体描述等。

## **谢 辞**

在进行毕业设计中，由老师辅导，进行毕业作品设计和完善，在大学时候，老师的教导和指导练习代码，有了一点基础，自己有空再看看资料，学习编程，到了企业公司后，老师按照方向专一指导学习和培训，再通过项目的指导完成学习，感谢老师的认真教导，让我能学会许多知识，有了很多练习的机会，更加完善自己的编程技术，经过自己毕业作品，毕设的设计，更加锻炼自己的能力，也感谢指导老师能帮我检查作品，改善和提出问题参考文献

（参考文献书写注意以下几点：1 ***参考文献正文***内容用***5号宋体字***，书写格式按照国家标准GB7714-87规定，或者按照学生***毕业设计手册*** 上关于参考文献的规定。 2 华东交通大学毕业设计要求参考文献数量在***10篇以上***，其中至少由***2篇以上英文参考文献***。）

参考文献是毕业设计中引用文献出处的目录表。参考文献一律放在文后，书写格式要按国家标准GB7714－87规定。

a. 专著、论文集、学位论文、报告

[序号] 著者. 文献题名[文献类型标识]. 出版地：出版者，出版年. 起止页码(任选).

[1] 刘国钧，陈绍业，王凤翥. 图书馆目录［M］. 北京：高等教育出版社，1957. 15-18.

[2] 辛希孟. 信息技术与信息服务国际研讨会论文集：A集［C］. 北京：中国社会科学出版社，1994.

[3] 张筑生. 微分半动力系统的不变集［D］. 北京：北京大学数学系数学研究所，1983.

[4] 冯西桥. 核反应堆压力管道与压力容器的LBB分析［R］. 北京：清华大学核能技术设计研究院，1997.

b. 期刊文章

[序号] 著者. 文献题名［J］. 刊名，年，卷(期)：起止页码.

[5] 何龄修. 读顾城《南明史》［J］. 中国史研究，1998，(3)：167-173.

[6] 金显贺，王昌长，王忠东，等. 一种用于在线检测局部放电的数字滤波技术［J］.清华大学学报(自然科学版)，1993，33(4)：62-67.

c. 电子文献

[序号] 著者. 电子文献题名［文献类型标识/载体类型标识］. 电子文献的出处或可获得地址，发表或更新日期/引用日期(任选).

[7]郭路. XML数据传输的安全加密［EB/OL］. http://www-900.ibm.com/developerWorks/cn/xml/ xmlb2b/index5.shtml, 2001-06/2001-10-04.

注1：参考文献类型

参考文献类型 专著 论文集 报纸文章 期刊文章 学位论文 报告 标准 专利

文献类型标识 M C N J D R S P

对于其他未说明的文献类型，建议采用单字母“Z”。

注2：电子文献类型

电子参考文献类型 数据库 计算机程序 电子公告

电子文献类型标识 DB CP EB

注3：电子载体类型

电子文献载体类型 磁带 磁盘 光盘 联机网络

电子载体类型标识 MT DK CD OL

如：[DB/OL]—联机网上数据库、[DB/MT]—磁带数据库、[M/CD]—光盘图书

[CP/DK]— 磁盘软件、[J/OL]— 网上期刊、[EB/OL]— 网上电子公告

在正文中引用参考文献：在引用处最后一个字的右上角，用方括号标明此序号。

如，“XML秘钥信息服务规范[1]”、“CAST对称秘钥算法[2],[4]”。

附录A 外文翻译—原文部分

此处注明原文的出处

附录B 外文翻译—译文部分

附录C 软件使用说明书

使用vscode软件打开代码，安装node和npm，使用命令启动运行程序，打开mongodb数据库服务，打开node程序。

进行程序，首先到达登录页面，/login地址，进行账号登录，如果没有账号，可以先注册，然后登录，自动跳到主页面/home，主页面是spa单页面应用，有不同的模块，飞机票，火车票，度假，团购，本地人，门票等，可以打开不同子页面，进行相关的操作，点击顶部的订单，可以查看所有的消费记录附录D 主要源代码

import React, { Component } from "react";

import { Input, Image, Button } from "antd";

// import { BrowserRouter as Router, NavLink as Link , Route, Switch } from 'react-router-dom'

import {

HashRouter as Router,

NavLink as Link,

Route,

Switch,

Redirect,

} from "react-router-dom";

import { withRouter } from "react-router";

import "./index.less";

import Head from "../../component/head";

import LeftFixed from "../../component/leftFixed";

import Main from "../../component/main";

import PlaneTicket from "../../component/planeTicket";

import Hotel from "../../component/hotel";

import TrainTickets from "../../component/trainTickets";

import Holiday from "../../component/holiday";

import GroupBuying from "../../component/groupBuying";

import Tickets from "../../component/tickets";

import Local from "../../component/local";

const { Search } = Input;

class Home extends Component {

constructor(props) {

super(props);

this.state = {

componentNum: "",

};

this.onSearch = this.onSearch.bind(this);

}

componentDidMount() {

console.log(this.props.location.pathname);

let pathName = this.props.location.pathname;

let index = "0";

if (pathName == "/main") {

index = "0";

}

if (pathName == "/planeTicket") {

index = "1";

}

if (pathName == "/hotel") {

index = "2";

}

if (pathName == "/trainTickets") {

index = "3";

}

if (pathName == "/holiday") {

index = "4";

}

if (pathName == "/groupBuying") {

index = "5";

}

if (pathName == "/tickets") {

index = "6";

}

if (pathName == "/local") {

index = "7";

}

this.setState({

componentNum: index,

});

}

onSearch() {}

toHomeChildRouter(index) {

this.setState({

componentNum: index,

});

if (index == 0) {

this.props.history.replace("/main");

}

if (index == 1) {

this.props.history.replace("/planeTicket");

}

if (index == 2) {

this.props.history.replace("/hotel");

}

if (index == 3) {

this.props.history.replace("/trainTickets");

}

if (index == 4) {

this.props.history.replace("/holiday");

}

if (index == 5) {

this.props.history.replace("/groupBuying");

}

if (index == 6) {

this.props.history.replace("/tickets");

}

if (index == 7) {

this.props.history.replace("/local");

}

}

render() {

return (

<div className="home">

<Head />

{this.state.componentNum == 0 ? (

<div className="goWhereInput">

<div className="leftImg">

<Image

style={{ width: 100, height: 40 }}

src="https://gw.alipayobjects.com/zos/antfincdn/aPkFc8Sj7n/method-draw-image.svg"

/>

</div>

<div className="input">

<Search

placeholder="搜索目的地"

onSearch={this.onSearch}

enterButton

/>

</div>

<div className="telePeople">

<Image

style={{ width: 100, height: 40 }}

className="img"

src="https://gw.alipayobjects.com/zos/antfincdn/aPkFc8Sj7n/method-draw-image.svg"

/>

<div className="content">

<p>

<span>国内客服</span>95117

</p>

<p>

<span>国外客服</span>1234567

</p>

</div>

</div>

</div>

) : (

""

)}

<div className="ComponentRouter">

<Button

type="text"

size="large"

className={this.state.componentNum == 0 ? "clickEd" : "Noclick"}

onClick={this.toHomeChildRouter.bind(this, "0")}

>

首页

</Button>

<Button

type="text"

size="large"

className={this.state.componentNum == 1 ? "clickEd" : "Noclick"}

onClick={this.toHomeChildRouter.bind(this, "1")}

>

机票

</Button>

<Button

type="text"

size="large"

className={this.state.componentNum == 2 ? "clickEd" : "Noclick"}

onClick={this.toHomeChildRouter.bind(this, "2")}

>

酒店

</Button>

<Button

type="text"

size="large"

className={this.state.componentNum == 3 ? "clickEd" : "Noclick"}

onClick={this.toHomeChildRouter.bind(this, "3")}

>

火车票

</Button>

<Button

type="text"

size="large"

className={this.state.componentNum == 4 ? "clickEd" : "Noclick"}

onClick={this.toHomeChildRouter.bind(this, "4")}

>

度假

</Button>

<Button

type="text"

size="large"

className={this.state.componentNum == 5 ? "clickEd" : "Noclick"}

onClick={this.toHomeChildRouter.bind(this, "5")}

>

团购

</Button>

<Button

type="text"

size="large"

className={this.state.componentNum == 6 ? "clickEd" : "Noclick"}

onClick={this.toHomeChildRouter.bind(this, "6")}

>

门票

</Button>

<Button

type="text"

size="large"

className={this.state.componentNum == 7 ? "clickEd" : "Noclick"}

onClick={this.toHomeChildRouter.bind(this, "7")}

>

当地人

</Button>

</div>

<Router>

<Switch>

<Route path="/main" component={Main} />

<Route path="/planeTicket" component={PlaneTicket} />

<Route path="/hotel" component={Hotel} />

<Route path="/trainTickets" component={TrainTickets} />

<Route path="/holiday" component={Holiday} />

<Route path="/groupBuying" component={GroupBuying} />

<Route path="/tickets" component={Tickets} />

<Route path="/local" component={Local} />

<Route path="/" exact component={Main} />

</Switch>

</Router>

<div className="fixedFixed">

<LeftFixed />

</div>

</div>

);

}

}

export default withRouter(Home);

登录

import React, { Component } from "react";

import SmallLogin from "../../component/login";

import LoginHead from "../../component/login/loginHead";

import "./index.less";

import { Image, Select, Form, Button, Input } from "antd";

import VerficationCode from "../../component/login/verificationCode";

const { Option } = Select;

class Home extends Component {

constructor(props) {

super(props);

this.verfication = React.createRef();

this.state = {

loginOrRegister: true,

};

this.checkLoginRegister = this.checkLoginRegister.bind(this);

this.onFinish = this.onFinish.bind(this);

this.onFinishFailed = this.onFinishFailed.bind(this);

}

checkLoginRegister(data) {

this.setState({

loginOrRegister: data,

});

}

onFinish(e) {

console.log(this.verfication.current.state);

console.log(e);

}

onFinishFailed() {}

render() {

return (

<div className="loginMain">

<div className="head">

<LoginHead checkLoginRegister={this.checkLoginRegister} />

</div>

<div className="loginCenter">

<div className="logImg">

<Image

style={{ width: 150, height: 50 }}

src="https://gw.alipayobjects.com/zos/antfincdn/aPkFc8Sj7n/method-draw-image.svg"

/>

{this.state.loginOrRegister ? (

<span>账号登录</span>

) : (

<span>账号注册</span>

)}

</div>

{this.state.loginOrRegister ? (

<div className="CenterMain">

<div className="MainLeft">

<Image

width={200}

src="https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ.png"

/>

</div>

<SmallLogin />

</div>

) : (

<>

<Form

labelCol={{

span: 4,

}}

wrapperCol={{

span: 14,

}}

layout="horizontal"

onFinish={this.onFinish}

onFinishFailed={this.onFinishFailed}

>

<Form.Item

label="国家/地区"

name="country"

style={{ width: 500 }}

rules={[

{

required: true,

message: "Please input your telephone!",

},

]}

>

<Select

showSearch

style={{ width: 200 }}

placeholder="Select a person"

optionFilterProp="children"

filterOption={(input, option) =>

option.children

.toLowerCase()

.indexOf(input.toLowerCase()) >= 0

}

>

<Option value="jack">Jack</Option>

<Option value="lucy">Lucy</Option>

<Option value="tom">Tom</Option>

</Select>

</Form.Item>

<Form.Item

name="telephone"

label="手机号码"

style={{ width: 500 }}

rules={[

{

required: true,

message: "Please input your telephone!",

},

]}

>

<Input />

</Form.Item>

<Form.Item

name="password"

label="登录密码"

style={{ width: 500 }}

rules={[

{

required: true,

message: "Please input your password!",

},

]}

hasFeedback

>

<Input.Password />

</Form.Item>

<Form.Item name="Code" label="验证码" style={{ width: 500 }}>

<VerficationCode ref={this.verfication} />

</Form.Item>

<Form.Item style={{}}>

<Button type="primary" htmlType="submit">

同意协议并注册

</Button>

</Form.Item>

</Form>

</>

)}

</div>

</div>

);

}

}

export default Home;

附录E 相关图表