**BrandiTravel网站概要设计**

**V3.0**

**三峡大学 数字媒体技术专业**

评审日期： 2019 年 11 月 20 日

目录

[1. 导言 1](#_Toc25179374)

[1.1 编写目的 1](#_Toc25179375)

[1.2 范围 1](#_Toc25179376)

[1.3 引用标准 1](#_Toc25179377)

[1.4 参考资料 1](#_Toc25179378)

[1.5 版本更新信息 2](#_Toc25179379)

[2. 项目设计原则简介 2](#_Toc25179400)

[3. 结构体系设计 3](#_Toc25179401)

[3.1 表现层 5](#_Toc25179402)

[3.2 控制层 7](#_Toc25179403)

[3.3 业务逻辑数据访问层 7](#_Toc25179404)

[3.4 接口设计 9](#_Toc25179405)

[4. 功能模块设计 9](#_Toc25179406)

[4.1 功能模块设计总述 9](#_Toc25179407)

[4.2 前台基本业务模块设计 11](#_Toc25179408)

[4.2.1 模块CM1：登录系统 11](#_Toc25179409)

[4.2.2 模块CM2：注册模块 12](#_Toc25179410)

[4.2.3 模块CM3：旅游路线展示 12](#_Toc25179411)

[4.2.4 模块CM4：旅游路线类型展示 13](#_Toc25179412)

[4.2.5 模块CM5：查看旅游路线详情 14](#_Toc25179413)

[4.2.6 模块CM6：团购旅游路线展示 15](#_Toc25179414)

[4.2.7 模块CM7：购物车功能 16](#_Toc25179415)

[4.2.8 模块CM8：填写出游人员信息 17](#_Toc25179416)

[4.2.9 模块CM9：查看订单功能 19](#_Toc25179417)

[4.2.10 模块CM10：确认订单功能 19](#_Toc25179418)

[4.2.11 模块CM11：删除订单功能 20](#_Toc25179419)

[4.2.12 模块CM12：支付订单功能 21](#_Toc25179420)

[4.3 后台管理模块设计 22](#_Toc25179421)

[4.3.1 模块AM1：管理员登录 22](#_Toc25179422)

[4.3.2 模块AM2：路线类型管理 23](#_Toc25179423)

[4.3.3 模块AM3：旅游路线管理 24](#_Toc25179424)

[4.3.4 模块AM4：团购信息管理 25](#_Toc25179425)

[5. 数据库设计 26](#_Toc25179426)

[5.1 数据库种类特点 26](#_Toc25179427)

[5.2 数据库逻辑结构 27](#_Toc25179428)

[5.3 物理结构设计 28](#_Toc25179429)

[5.3.1 表1：Customer表 29](#_Toc25179430)

[5.3.2 表2：LineType表 29](#_Toc25179431)

[5.3.3 表3：Line表 29](#_Toc25179432)

[5.3.4 表4：Car表 30](#_Toc25179433)

[5.3.5 表5：Picture表 30](#_Toc25179434)

[5.3.6 表6：OrderDetail表 30](#_Toc25179435)

[5.3.7 表7：Tourist表 31](#_Toc25179436)

[5.3.8 表8：OT\_Detail表 31](#_Toc25179437)

[5.3.9 表8：Manager表 31](#_Toc25179447)

[6. 界面设计 32](#_Toc25179448)

[6.1 首页设计 32](#_Toc25179449)

[6.2 用户注册/登录页面设计 33](#_Toc25179450)

[6.3 BrandiTravel旅游页面设计 34](#_Toc25179451)

[6.3.1 旅游路线概况页面设计 34](#_Toc25179452)

[6.3.2 旅游路线类型页面设计 35](#_Toc25179453)

[6.3.3 旅游路线详情页面设计 36](#_Toc25179454)

[6.3.4 团购旅游路线页面设计 37](#_Toc25179455)

[6.4 BrandiTravel购物车页面设计 37](#_Toc25179456)

[6.5 BrandiTravel订单页面设计 38](#_Toc25179457)

[6.5.1 订单界面 38](#_Toc25179458)

[6.5.2 填写出游人员信息页面 39](#_Toc25179459)

[6.6 后台主页面设计 40](#_Toc25179460)

[6.7 后台管理员登录页面设计 41](#_Toc25179461)

[6.8 旅游路线管理页面设计 41](#_Toc25179462)

[6.9 旅游路线类型管理页面设计 42](#_Toc25179463)

[6.10 团购旅游路线管理页面设计 44](#_Toc25179471)

# 导言

## 编写目的

该文档根据BrandiTravel的功能和性能，阐述了BrandiTravel站的概要设计，包括框架设计，功能模块设计，数据库设计，界面设计等部分。

本文档的预期读者包括：

* 设计开发人员
* 项目管理人员
* 测试人员
* 用户

## 范围

该文档主要目的是为了解决项目系统去解决“该怎么去做”的问题。包括了对用户提出的问题解决方案，自身项目可能导致的问题解决方案，以及未来可预期的问题解决方案。

## 引用标准

[1]《SPM 课程网站概要设计》文档 北京邮电大学

[2]《BrandiTravel网站概要设计》文档 三峡大学

## 参考资料

[1] 软件工程：实践者的研究方法（原书第八版·本科教学版）/（美）罗杰 S.普莱斯曼（Roger S.Pressman），（美）布鲁斯 R.马克西姆（Bruce R.Maxim）著；郑仁杰等译.—北京：机械工业出版社；

[2] UML2面向对象的分析与设计/谭火彬著.—北京：清华大学出版社；

[3] 人月神话（40周年中文纪念表）/(美)布鲁克斯（Brooks,F.P.）著；UML China翻译组，汪颖译.—北京：清华大学出版社。

## 版本更新信息

本文档更新记录如表 1‑1所示。

表 1‑1版本更新信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修改编号 | 修改日期 | 修改后版本 | 修改位置 | 修改内容概述 |
| 001 | 2019.10.31 | 1.0 | 全部 | 整体文档初始发布 |
| 002 | 2019.11.10 | 2.0 | 局部 | 部分内容修改 |
| 003 | 2019.11.20 | 3.0 | 局部 | 完善修改 |

# 项目设计原则简介

随着人类生活质量的不断提高，人类对精神生活质量的追求也越来越高，旅游也成为了人类满足物质生活和精神生活的需要。但有些人经常在出游时，为选择线路而举棋不定，不知道哪条线路的性价比最高。是一款能够方便、快速地为出游人员选择最适合的线路的网站在整个系统设计的过程中遵循以下设计原则：

1.实用性：实用性是系统的主要设计原则，系统设计必须最大可能地满足用户的需求，做到操作方便、界面友好、可即时更新，能够及时显示如今可执行的旅游路线

2.先进性：信息技术发展迅速，系统设计尽可能采用先进的技术标准和技术方法。

3.以用户为中心的处理：个性化服务充分体现了这一点，根据用户当前意愿为主，配置页面功能布局及展现内容，贴合用户操作。

4.使用便捷。系统要有设计良好的人机交互界面，即使系统的操作界面简单易用，又能具有较强的适用性，满足不同计算机使用水平的用户使用。

5.灵活和易维护：采用开放的体系架构，基于开放源代码的技术框架和数据库系统，使用高效率的开源和免费开发工具，具备完整的文档说明。在维护方面，主要考虑两个层面，一是对于开发人员来讲，系统编码容易调整，可适应需求的变化和调整；二是对于系统管理维护人员来说，能够对系统进行便捷的维护和管理。

6.安全可靠：选择安全可靠的软硬件运行平台，并在系统设计和实现的时候关注系统的安全控制和执行效率，提供相应的安全防护功能，保证系统具有较高的安全性和可靠性。安全性方面，要考虑系统的安全、数据管理的安全、网络安全。保证用户权限、数据安全和系统的稳定性。

7.单一职责原则：我们系统在面向对象设计部分采取单一职责原则，其核心思想为：一个类，最好只做一件事，只有一个引起它的变化。单一职责原则可以看做是低耦合、高内聚在面向对象原则上的引申，将职责定义为引起变化的原因，以提高内聚性来减少引起变化的原因。从而最终提高我们系统的可修改性和可维护性。

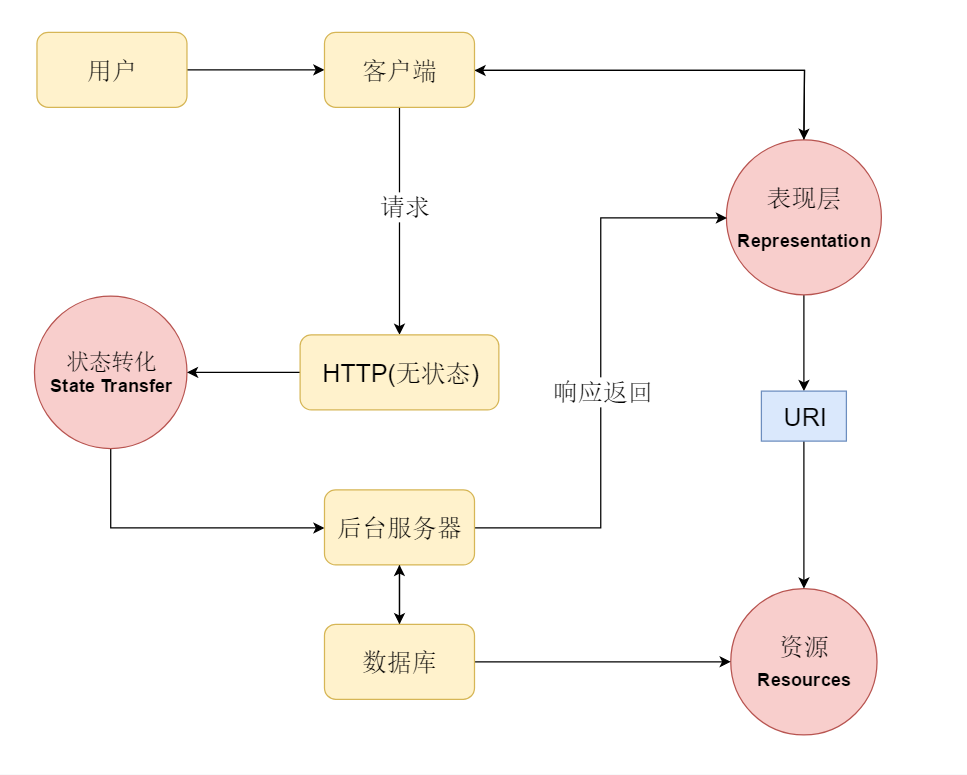
# 结构体系设计

BrandiTravel系统本着软件开发的设计原则，采用浏览器/服务器（B/S）的体系结构。为了满足网站后台的增删改查和平台用户创建信息与信息收发的基本操作，在软件架构上，采用三层体系结构：控制层，表现层，业务逻辑层和数据访问层；在设计实现上，我们采用MVC的设计模式：Model 模型层、View 视图层、Controller 控制层；在体系架构上，BrandiTravel系统选择用Rest API架构。

Rest API（Representational State Transfer Application Programming Interface）框架集由**Resources（资源）**，**Representation（表现层）**、**State Transfer（状态转换）**三个开源框架整合而成，大多数情况下，当你在使用浏览器的点点看看的时候，其实只用到HTTP的GET方法。GET方法是在向因特网请求资源的时候才会用到的。当提交一个表单时，就会经常用到POST方法来回传数据到网站上。至于其他的几种方法，某些浏览器可能根本就没有去完全实现它们。但是，如果是供我们使用的话，就没什么问题。问题是我们有很多要选择去帮助描述这四大行为的HTTP方法，我们将会用到那些已经知道如何去使用这些不同的HTTP方法的客户端类库。

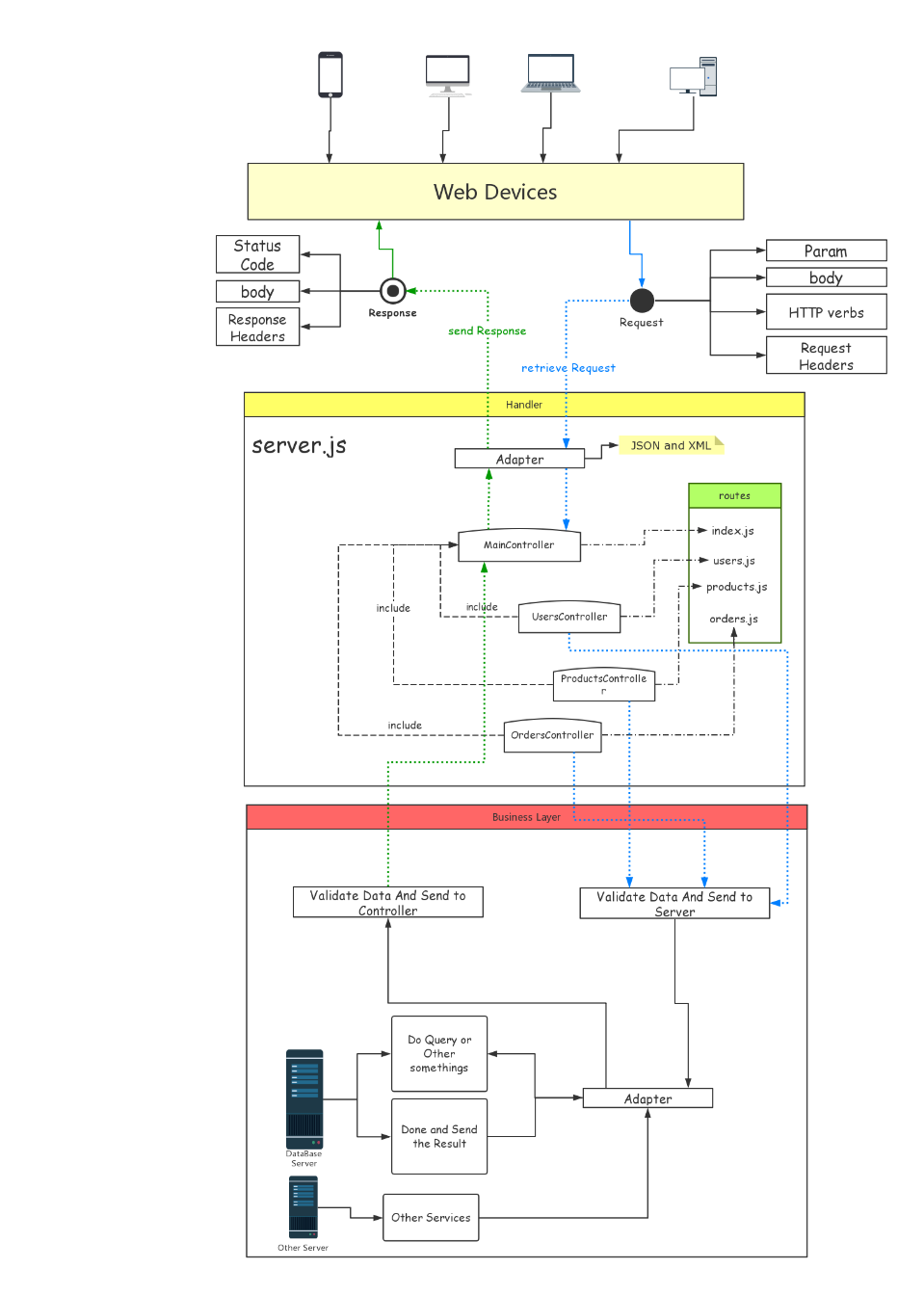
Rest API框架关系如图 3‑1所示：

图 3‑1Rest API框架关系示意图



Rest API架构运行流程如图 3‑2所示：

图 3‑2 Rest API架构运行流程



对于Rest API的三个不同开源框架分别介绍为：

**资源（Resources）**

REST的名称"表现层状态转化"中，省略了主语。"表现层"其实指的是"资源"（Resources）的"表现层"。所谓"资源"，就是网络上的一个实体，或者说是网络上的一个具体信息。它可以是一段文本、一张图片、一首歌曲、一种服务，总之就是一个具体的实在。你可以用一个URI（统一资源定位符）指向它，每种资源对应一个特定的URI。要获取这个资源，访问它的URI就可以，因此URI就成了每一个资源的地址或独一无二的识别符。

所谓"上网"，就是与互联网上一系列的"资源"互动，调用它的URI。

**表现层（Representation）**

"资源"是一种信息实体，它可以有多种外在表现形式。我们把"资源"具体呈现出来的形式，叫做它的"表现层"（Representation）。

比如，文本可以用txt格式表现，也可以用HTML格式、XML格式、JSON格式表现，甚至可以采用二进制格式；图片可以用JPG格式表现，也可以用PNG格式表现。URI只代表资源的实体，不代表它的形式。严格地说，有些网址最后的".html"后缀名是不必要的，因为这个后缀名表示格式，属于"表现层"范畴，而URI应该只代表"资源"的位置。它的具体表现形式，应该在HTTP请求的头信息中用Accept和Content-Type字段指定，这两个字段才是对"表现层"的描述。

**状态转化（State Transfer）**

访问一个网站，就代表了客户端和服务器的一个互动过程。在这个过程中，势必涉及到数据和状态的变化。互联网通信协议HTTP协议，是一个无状态协议。这意味着，所有的状态都保存在服务器端。因此，如果客户端想要操作服务器，必须通过某种手段，让服务器端发生"状态转化"（State Transfer）。而这种转化是建立在表现层之上的，所以就是"表现层状态转化"。

## 表现层

结合用户身份判定，用于展示系统的业务信息以及接受用户输入信息。将来自用户的信息发送到对应的业务层进行处理，并接受后台处理的结果， 并结合用户身份将其返回到前端进行展示，实现系统与用户的动态交互。

表现层功能包括：

* 根据用户身份信息进行识别登录，显示不同界面。
* 根据用户身份信息进行识别实用系统，进行不同的数据和逻辑交互，满足用户需求 。
* 通过页面控件，完成统一风格的页面部署，形成较好的页面风格。
* 通过URI连接资源，包括图片，PDF，视频播放插件等等。
* 对前端提交数据进行合规性校验，如登入时账号密码长度，类型，格式等等。
* 将页面请求发送到验证层，并接受验证层返回，将结果在前端展示。
* 提供过滤器，进行请求预处理。

网页视图组件：根据业务逻辑层的处理数据进行页面的展示。网页视图部分样例表间表 3‑2。

表 3‑2 网页视图部分样例表

|  |  |  |
| --- | --- | --- |
| 文件名称 | Html | 说明 |
| **用户注册** | test.html | 显示用户注册界面 |
| **网页主页** | home.html | 显示网页主页界面 |
| **路线详情** | Main.html | 显示路线详情界面 |
| **境外游路线** | Out.html | 显示境外游路线界面 |
| **购物车** | shopping.html | 显示购物车界面 |
| **填写订单** | The order.html | 显示填写订单界面 |
| **支付订单** | confirm order.html | 显示支付订单界面 |

## 控制层

根据不同的请求触发点，接收从表现层传输的用户请求信息，并将封装好的实体对象发送到对应的业务处理单元，同时接收业务逻辑层处理结果。从表现层得到的全部Request将接受到Server.js主控制单元当中，再根据不同的反馈结果，选择相应的子控制单元.js跳转，实现页面跳转与信息现实。

控制层功能包括：

* 接收并处理从表现层传入的各种输入以及输出各种异常提示信息或处理结果信息。
* 对于输入的数据进行数据校验，过滤非法数据。
* 向业务控制层发送处理请求。
* 接收业务逻辑层请求处理结果。
* 返回状态符，实现页面跳转。

控制包部分结构见表 3‑3。

表 3‑3控制部分结构表

|  |  |
| --- | --- |
| 控件名 | 说明 |
| MainController | 实现主页面页面跳转 |
| UserController | 实现用户界面页面跳转 |
| ProductController | 实现所有路线页面跳转 |
| OrdersController | 实现订单功能页面跳转 |

## 业务逻辑数据访问层

业务逻辑层为系统的核心层，提供了大量业务服务组件，负责处理控制层发送过来的业务数据，系统总绝大部分业务处理都在该层实现。该层实现了各种逻辑判断，即业务逻辑的封装，实现各种需求功能，并将处理后的数据传输给控制层，再由控制层进行简单处理。如果需要进行数据库操作，则调用数据访问层进行数据库相关操作。

业务逻辑层功能包含：

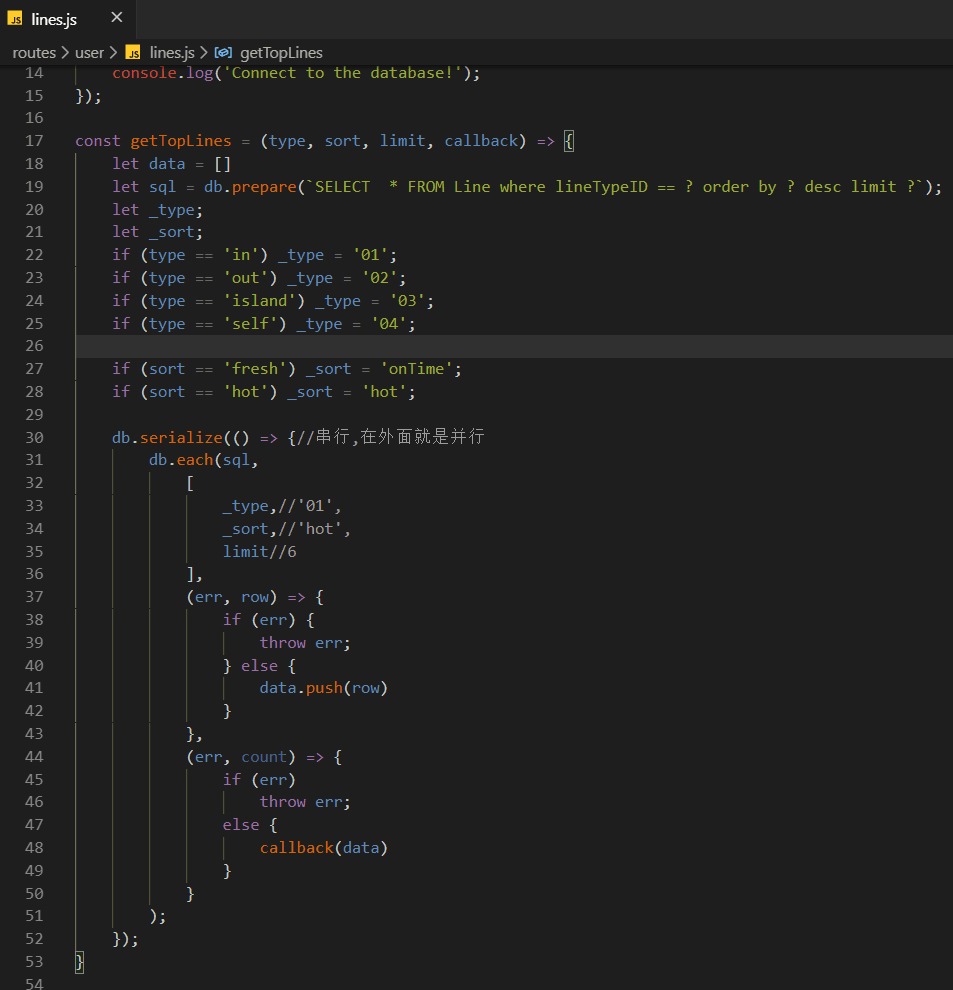
* 实现各种业务处理逻辑或处理算法，比如路线展示，类型分类。
* 向控制层返回处理数据信息。
* 向访问层发送数据操作的请求，进行对数据信息的增删改查操作。
* 作为控制层的服务层，提供接口供数据访问调用。

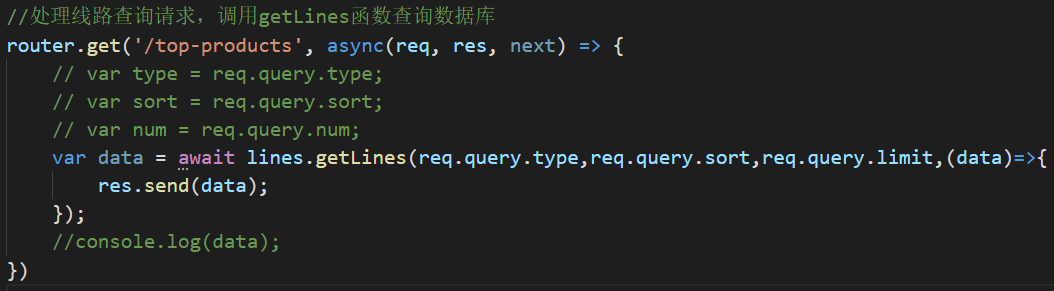
在BrandiTravel平台系统中，数据访问层负责数据的访问操作，例如和数据库交互与数据库进行连接交互。该层抽象和封装了所有对访问化存储介质的访问。数据访问层可以访问任何的数据，除了数据库，还包括缓存存数据等。在该项目框架的搭建中，是通过调用Line.js业务逻辑处理单元来访问数据库并进行相应返回。

数据访问层包含功能：

* 对数据库中的用户信息，路线信息，路线类型信息等数据进行增、删、改、查操作。
* 实现事务处理，保证数据读写正常。
* 实现意外错误操作的数据恢复。

由于使用的Rest API框架结构，Server.js当中的子控制单元的访问接口转移到了一个Line.js业务逻辑处理单元上，每一个子控制单元的.js的存在一个或多个代表对应功能的参数值，通过不同的参数值传值实现Line.js业务逻辑处理单元对数据库进行不同行为的访问，数据库响应并返回相应的访问值结果，再由适配器和部分控件传到控制层。可以回传给子控制单元，再由子控制单元单独传出，不再经由server.js主控制单元。减少了并行数据传递的压力。





## 接口设计

每一个对应的接口不由类传递，而是通过http协议传送请求命令进行单个或多个参数值传递。

|  |  |  |  |
| --- | --- | --- | --- |
| 1. HTTP verb | CRUD action | | Instance:  /products?lineID=01 |
| GET | **READ** | 获取线路详情。  **Status Code: 200 ok**  **Error Code: 4xx** | |
|  |  | Collection:  /top-products?type={type}&sort={sort}&order={order}&day={daynum}&onlygroup={onlygroup}&limit={num} | |
| GET | **READ** | 查询线路信息的有序集合。  参数说明：  type = {all,in,out,island,self}  sort = {hot, fresh}  order = {asc, desc}  daynum = {1,2,3},3表示多日游  onlygroup = {true,false}, 是只查询团购线路  num = {>1},查询前num条线路  **Status Code: 206 Partial Content** | |
|  |  | Instance:  /users/{userId}/addtocart?lineID= {lineID} | |
| POST | **CREATE** | 添加到购物车；  addCart是导向的购物车页面名  **Status Code: 201 ok** | |
|  |  | Instance:  /users/{userId}/addorder?lineID={lineID} | |
| POST | **CREATE** | 创建订单；  **Status Code: 201 ok** | |
|  |  | Collection：  /users/{userId}/orders | |
| GET | **READ** | 查询订单列表；  UserID是当前用户的一个标识  **Status Code: 200 ok** | |
|  |  | Instance:  /users/{userID}/deleteorder?lineID={lineID} | |
| DELETE | **DELETE** | 删除特定的一个购物车条目；  **Status Code: 204 ok** | |
|  |  | Instance:  /users/{userID}/deletecartitem?lineID={lineID} | |
| DELETE | **DELETE** | 删除特定的一个订单  **Status Code: 204 ok** | |

# 功能模块设计

## 功能模块设计总述

本系统为用户交互式管理平台，主要包括得模块为两部分：前台基本业务模块和后台管理模块，如图 4‑1所示。其中，图 4‑2是前台基本业务模块图，图 4‑3是后台管理模块图。

图 4‑1BrandiTravel主要模块图

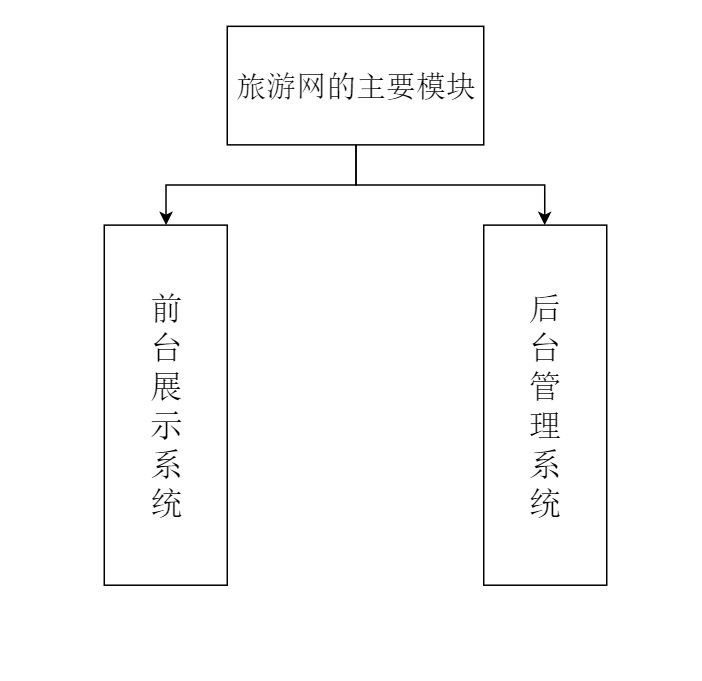


图 4‑2BrandiTravel后台管理模块图

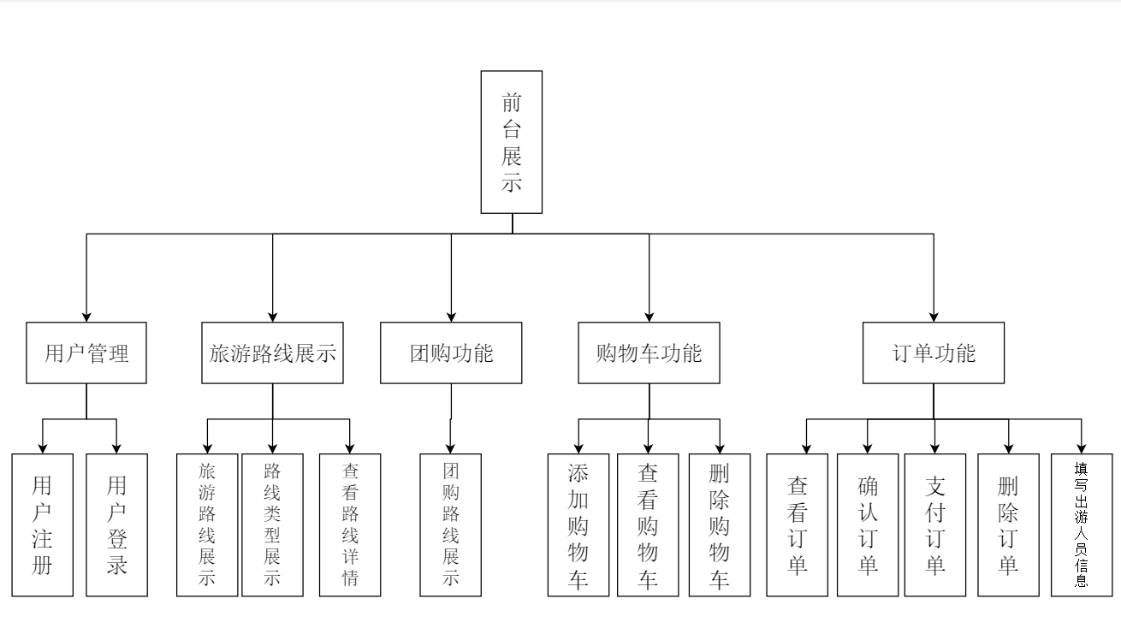
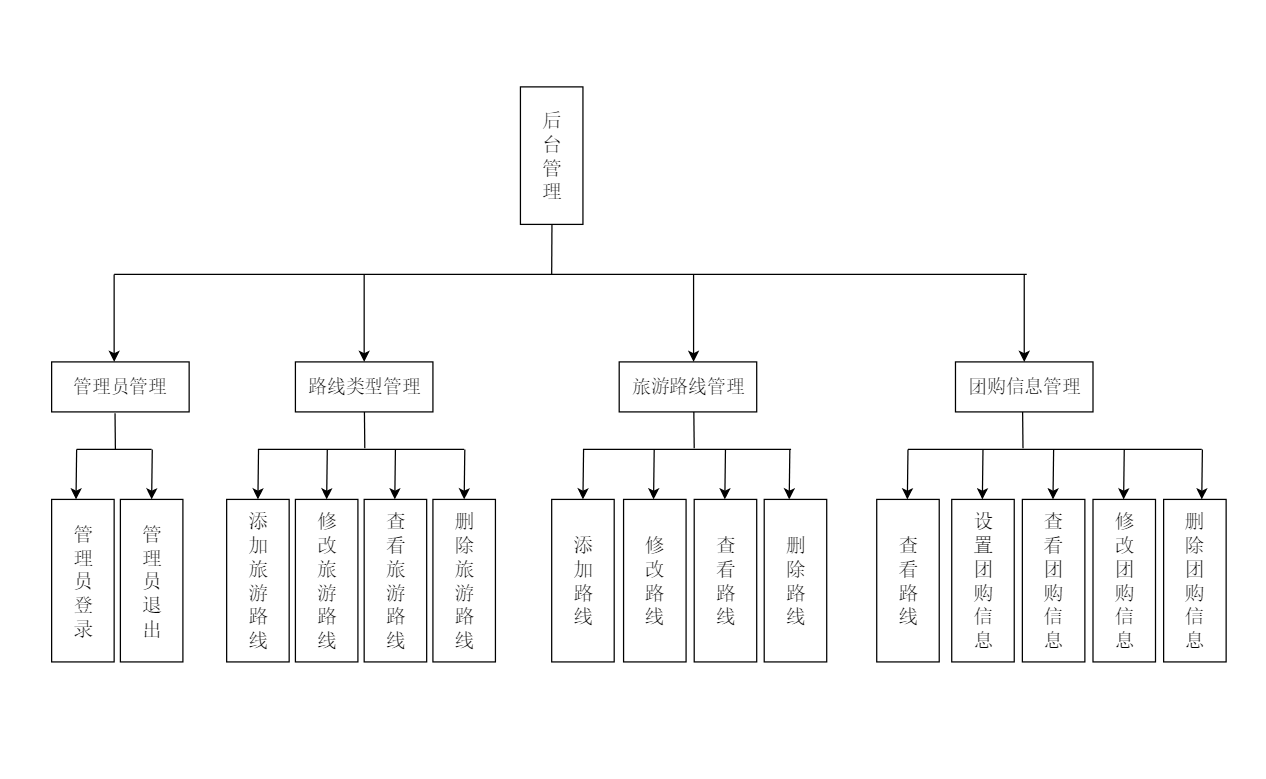


图 4‑3BrandiTravel后台管理模块图



## 前台基本业务模块设计

### 模块CM1：登录系统

**编号：**CM1；

**模块名称：**登录系统；

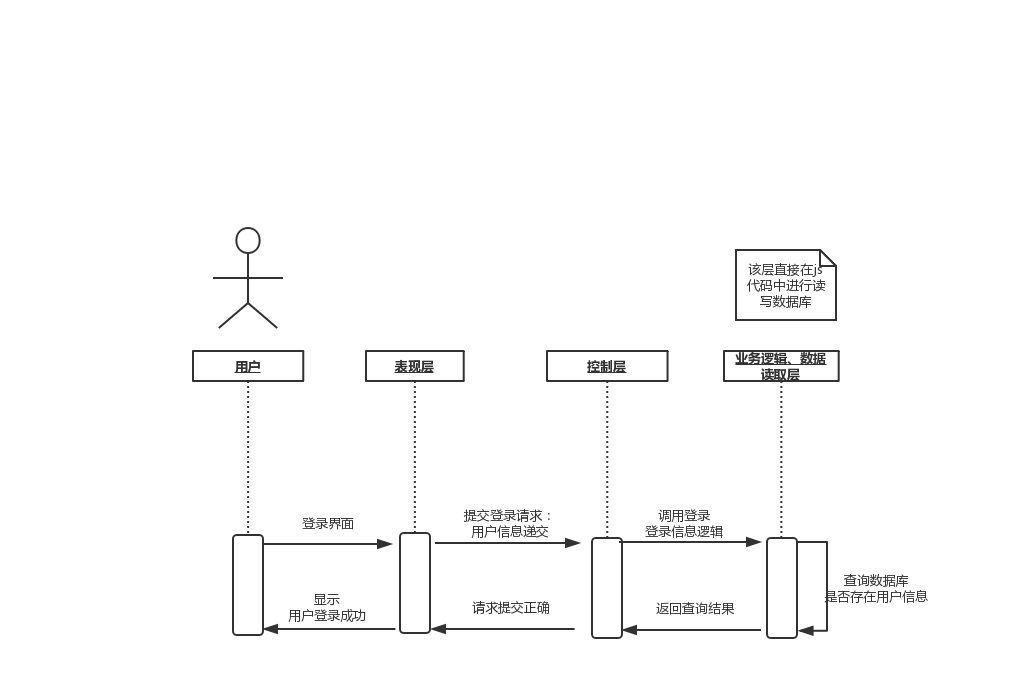
**功能简介：**本模块为系统登录模块，即会员登录系统的入口。在此模块中， 用户输入自己的注册邮箱和密码，系统在后台数据库进行查询操作后，返回布尔值，表示该输入是否正确，输入正确则进入系统，错误则对用户进行相应提示；

**输入：**注册邮箱，密码；

**输出：**用户是否登录成功；

**操作流程：**如图 4‑4所示。

图 4‑4登录系统操作流程



### 模块CM2：注册模块

**编号：**CM2；

**模块名称：**注册系统；

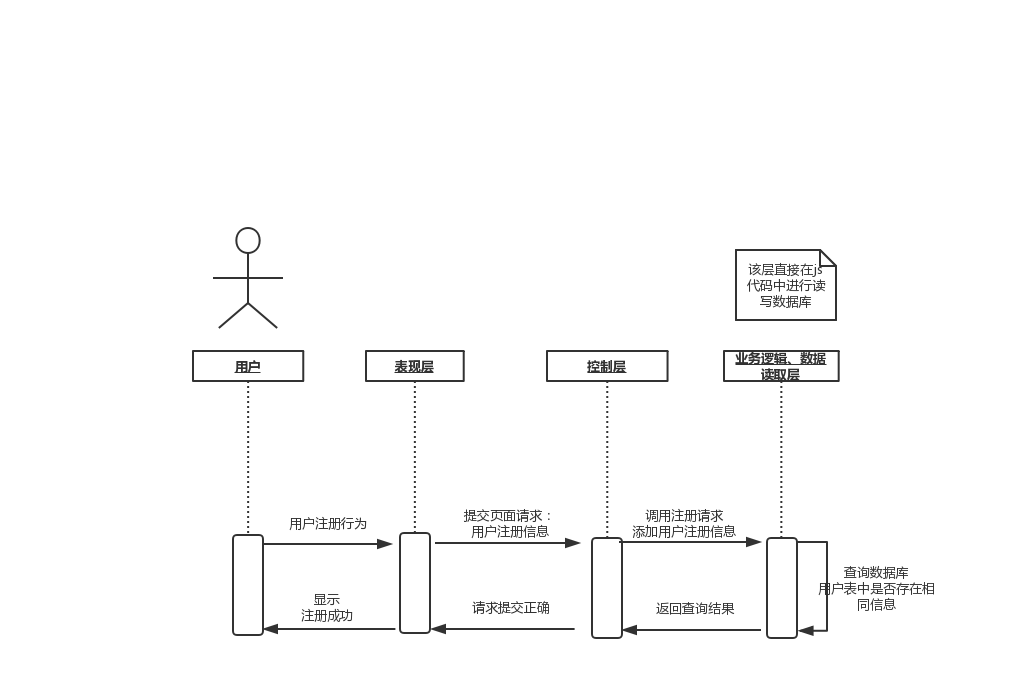
**功能简介：**本模块为系统注册模块。用户首次进入系统时。可通过本模块进行注册。在此模块中，系统显示注册界面，用户输入相关必要的身份信息，单击确定。若注册成功，系统将以注册邮箱作为账号，用户设定密码作为密码存入后台数据库。注册完成后，用户可使用注册成功的账号和密码登录系统；

**输入：**注册邮箱，密码，验证码等相关信息；

**输出：**用户是否注册成功；

**操作流程：**如图 4‑5所示。

图 4‑5注册系统操作流程



### 模块CM3：旅游路线展示

**编号：**CM3；

**模块名称：**旅游路线展示；

**功能简介：**用户登录网站后，可以直接进行旅游路线的大致浏览，其中可以对路线进行图片的参考，若对某一旅游路线感兴趣可以点击进入旅游路线详情。其中旅游路线展示包括了对旅游路线前5推荐，路线首页轮流播放；

而且页面的所有的图片均是可以点击进入的，点击后跳转页面到改旅游路线的详情界面。包括推荐的旅游路线，是通过后台对旅游路线的点击量，提交量进行计算，然后反馈到首页界面。

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图 4‑6所示。

图 4‑6旅游路线展示流程



### 模块CM4：旅游路线类型展示

**编号：**CM4；

**模块名称：**旅游路线类型展示；

**功能简介：**用户可以通过直接点击旅游路线类型，进入该界面，然后根据不同的旅游路线类型进行相应的选择。不同的旅游路线有着不同的类型，但是一条路线可以有多重类型，比如马尔代夫群岛旅游，既属于境外游也属于海岛游。不过针对的类型必然是不同的。

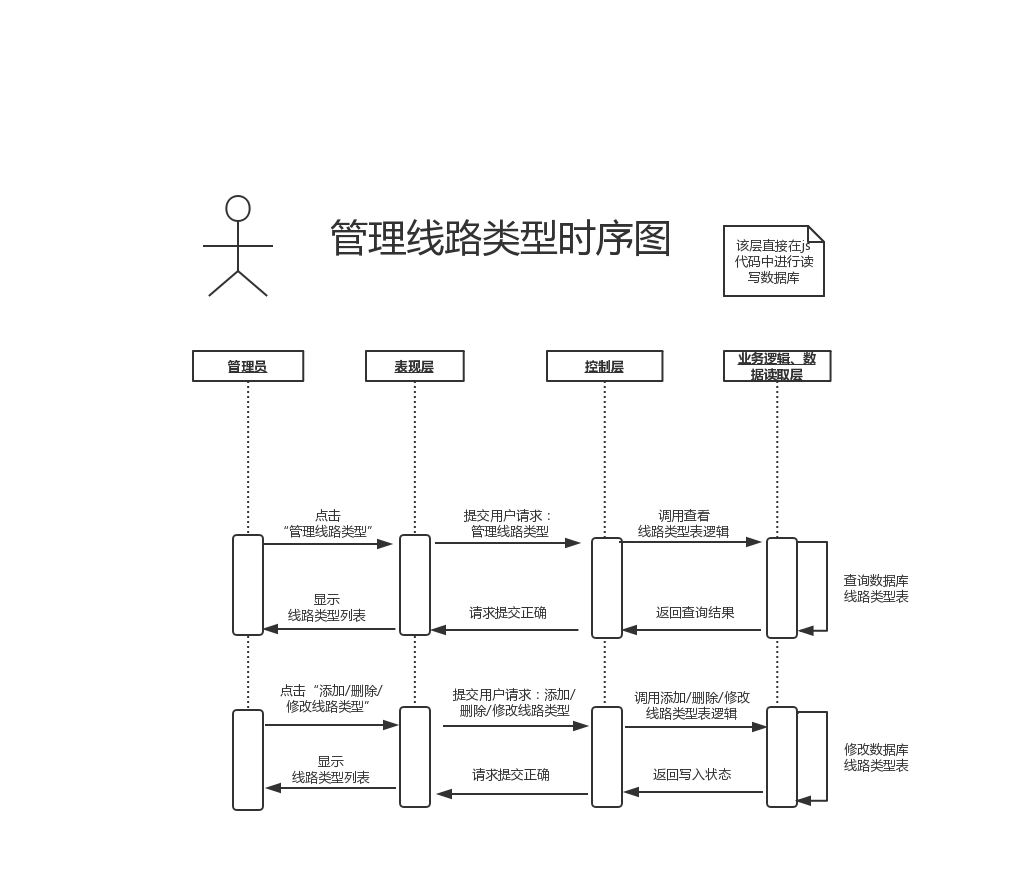
用户们可以在不同类型中进行选择，点击后就进入同类型的旅游路线页面查看。；

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图 4‑7所示。

图 4‑7旅游路线类型展示流程



### 模块CM5：查看旅游路线详情

**编号：**CM5；

**模块名称：**查看旅游路线详情；

**功能简介：**用户通过主页的旅游路线图片或者是从旅游路线类型中景点图片点击，都可以跳转到该路线的详情介绍，详情中包括了对路线的描述，景点简介图片，旅游路线乘坐的交通工具，路线的花费价格，路线天数以及推荐旅游人数。

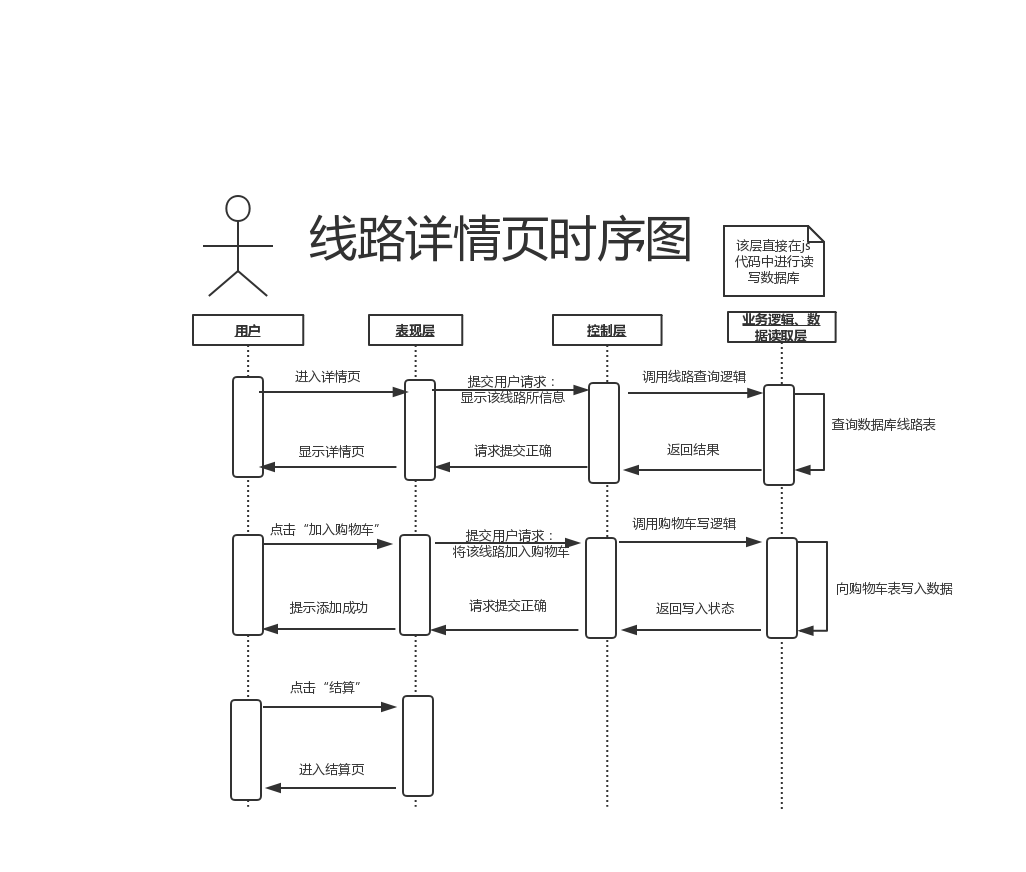
在用户针对详情的情况下可以进行添加至购物车操作，添加后进入购物车面板就可以查看改旅游路线商品，当然也可以直接在详情界面进行直接的支付操作，跳转到填写订单界面然后填写完毕提交支付即可。

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图 4‑8所示。

图 4‑8查看旅游路线详情流程



### 模块CM6：团购旅游路线展示

**编号：**CM6；

**模块名称：** 团购旅游路线展示

**功能简介：** 当用户们认为某一旅游路线出行人数过多，或是对某一路线单人出行费用过高，可以查看团购旅游路线，然后在其中找寻自己想要的路线，填写相应订单并提交。或是添置购物车中稍后处理也可以。

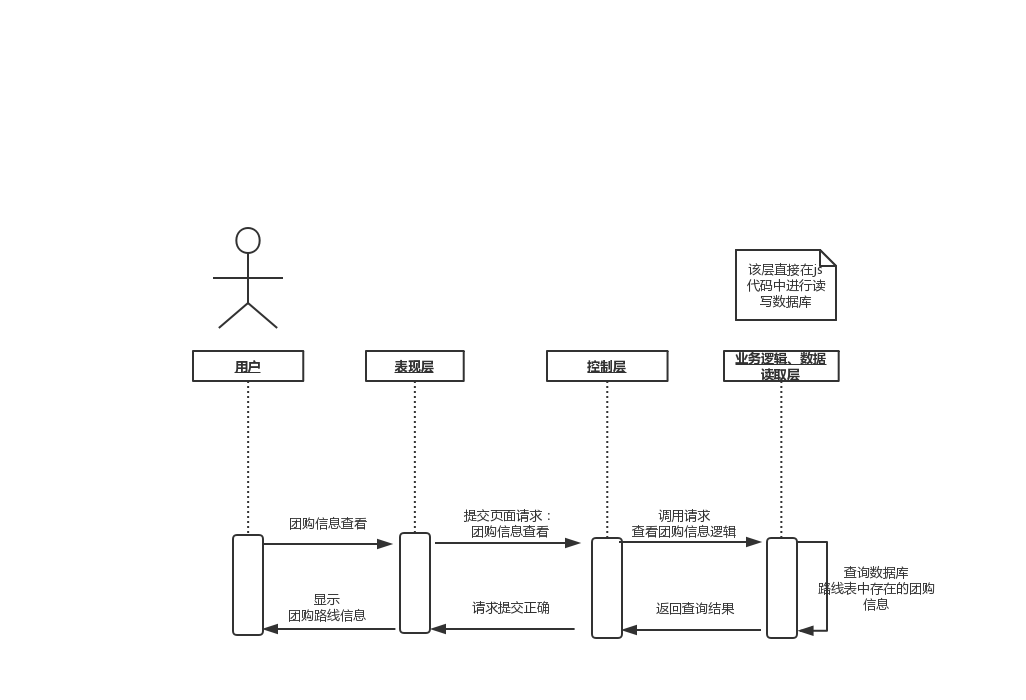
该功能存在部分不足在于用户查看的团购旅游路线是由管理员后天设置修改，并不是针对每一条路线，用户们也可能存在想要的路线并不存在在团购信息当中。

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图 4‑9所示。

图 4‑9团购路线查看操作流程



### 模块CM7：购物车功能

**编号：**CM7；

**模块名称：**购物车功能

**功能简介：** 用户对于中意的路线可以进行添加至购物车功能，方便用户对已选的路线商品进行更改或选择结账。也保留了用户们对相应路线选择等待的机会。

路线如果再原有界面被管理员进行修改过后，购物车中的路线商品也会自动跟随修改，并且修改的类型和价格也会随之改变。当原有商品路线下架了，购物车中商品则会提示存在的商品下架，自动移除。

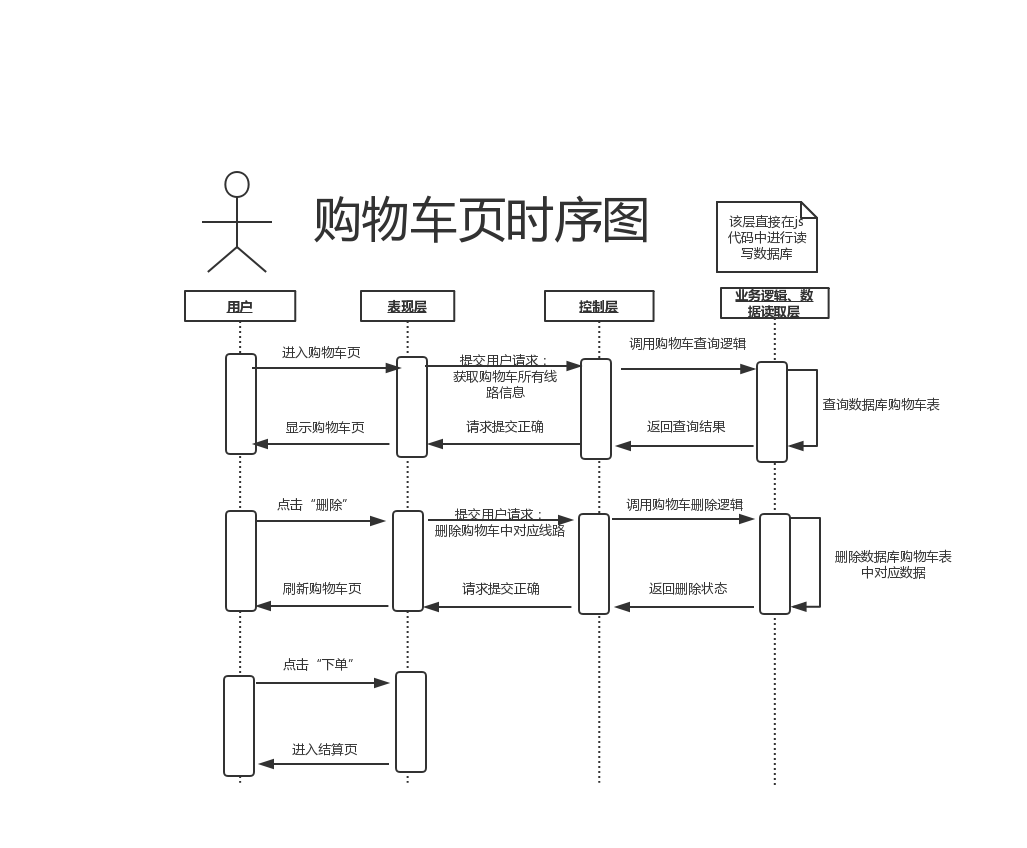
购物车中的商品路线可以进行两种操作，一个是移除，一个是结账处理。添加购物车商品路线则是在路线详情中用户自主添加。移除则将商品路线从购物车中执行栈中出栈操作。当选择结账时，购物车界面会跳转到订单界面，由用户先填写订单概况，提交订单并支付。

**输入：**鼠标点击事件；

**输出：** 页面刷新；

**操作流程：** 如图 4‑10所示

图 4‑10信息修改操作流程



### 模块CM8：填写出游人员信息

**编号：**CM8；

**模块名称：** 填写出游人员信息；

**功能简介：** 用户在提交相应订单之前进行的必要操作。填写出游人员的真实姓名，身份证号以及电话号码三个主要个人信息。以方便在出游临近阶段方便提醒并联系相应的出游人员。填写的出游人员可以不用是用户，但相关提示信息会反馈到用户账户上。填写的信息也会上传至服务器，并传递到相应路线景点的负责人那进行出游的核实。

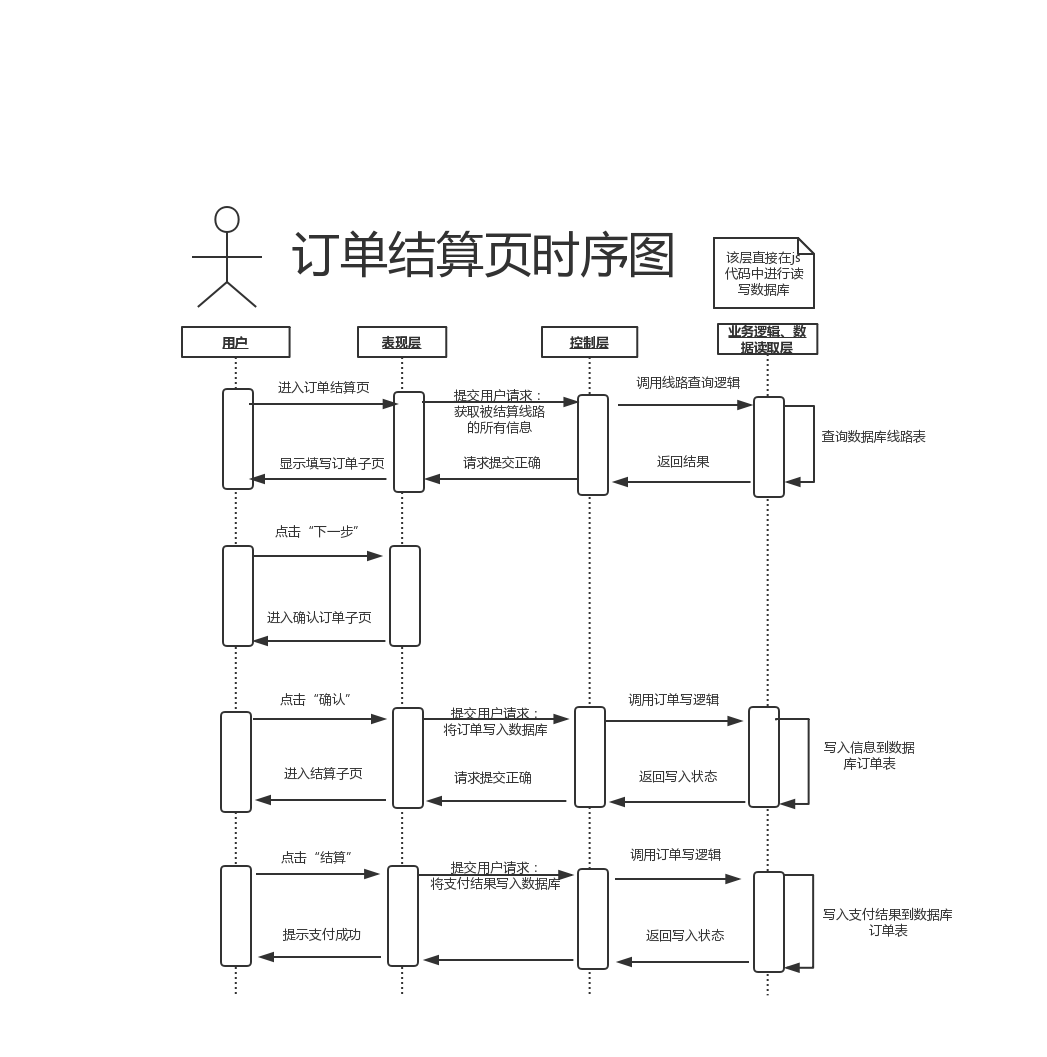
为确保出游人员的真实性，也会将其身份证号联系公安局档案，填写信息可以为多人，如果有想再添加多位出游人员，点击“添加出游人员”按钮即可。在出游人员信息统计填写完成后，点击下一步即可确认订单，提交并支付。

**输入：**文本输入；

**输出：**页面刷新；

**操作流程：**如图 4‑11所示。

图 4‑11填写出游人员信息操作流程



### 模块CM9：查看订单功能

**编号：**CM9；

**模块名称：**查看订单功能

**功能简介：**用户们在一次旅行路线结束后，当次的订单仍然会保留在其用户信息当中，成为历史订单，用户在后来可以再次对相关订单的查看或再一次旅游。历史订单方便用户查看上一次路线的情况，或者对该路线进行推荐。

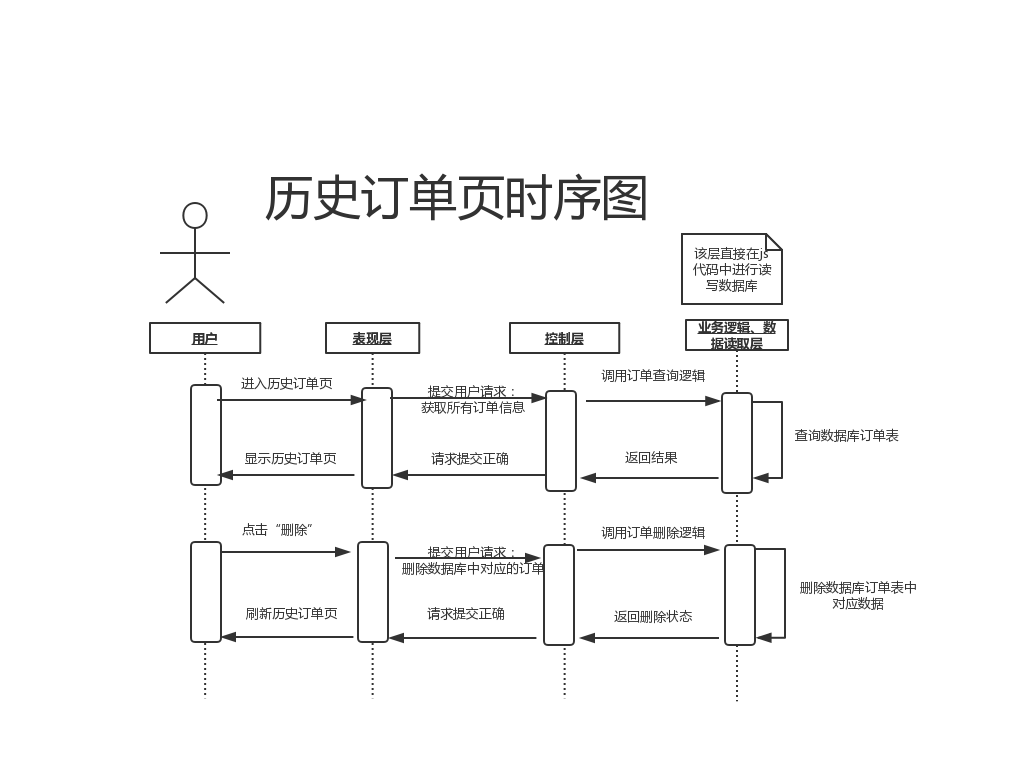
若用户由其他不可抗力出现出游时间未赶上。可以通过历史订单进行相应后续处理，包括后续开发功能的改签，退款等操作。

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图 4‑12所示。

图 4‑12查看订单操作流程



### 模块CM10：确认订单功能

**编号：**CM10

**模块名称：**确认订单功能

**功能简介：** 用户在填写完订单信息和出游人员信息后，会进行短信通知和验证码提示，确认用户信息核实，确认完毕后才能进行付款操作，

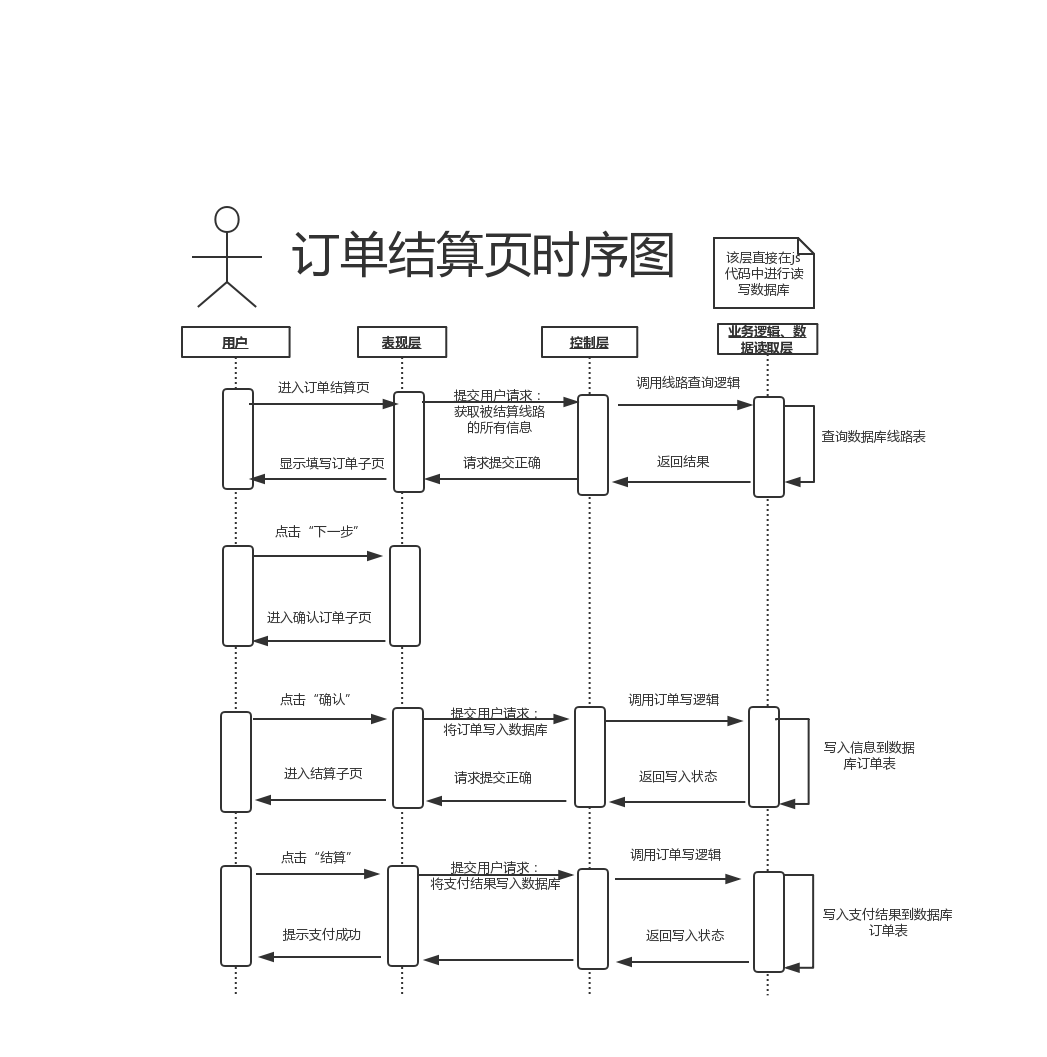
该操作与跳转至支付界面相链接。 所以该功能那个确保了整个用户信息的完整性与真实性，保障了不会出现用户误填产生的错误。

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图 4‑13所示。

图 4‑13确认订单操作流程



### 模块CM11：删除订单功能

**编号：**CM11

**模块名称：**删除订单功能

**功能简介：**用户对某一订单意愿的删除操作，对订单列表进行查看后，用户可自行对已有的订单进行删除操作处理。该操作也会进行后台数据库传递数据，然后从数据库信息中将相关订单信息从用户信息删除。

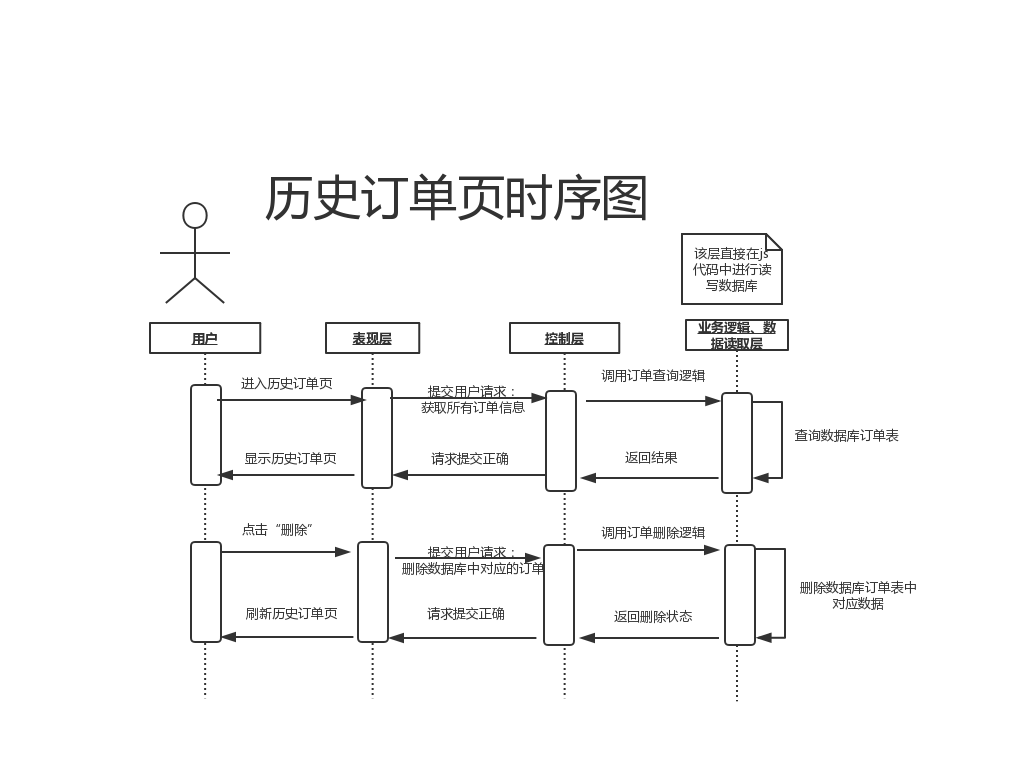
删除操作时不可逆的，用户对于订单的删除会清空相关的订单内容，包括填写的该路线出游人员信息，所以用户必须谨慎操作，确认删除。

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图 4‑14所示。

图 4‑14删除订单操作流程



### 模块CM12：支付订单功能

**编号：**CM12

**模块名称：**支付订单功能

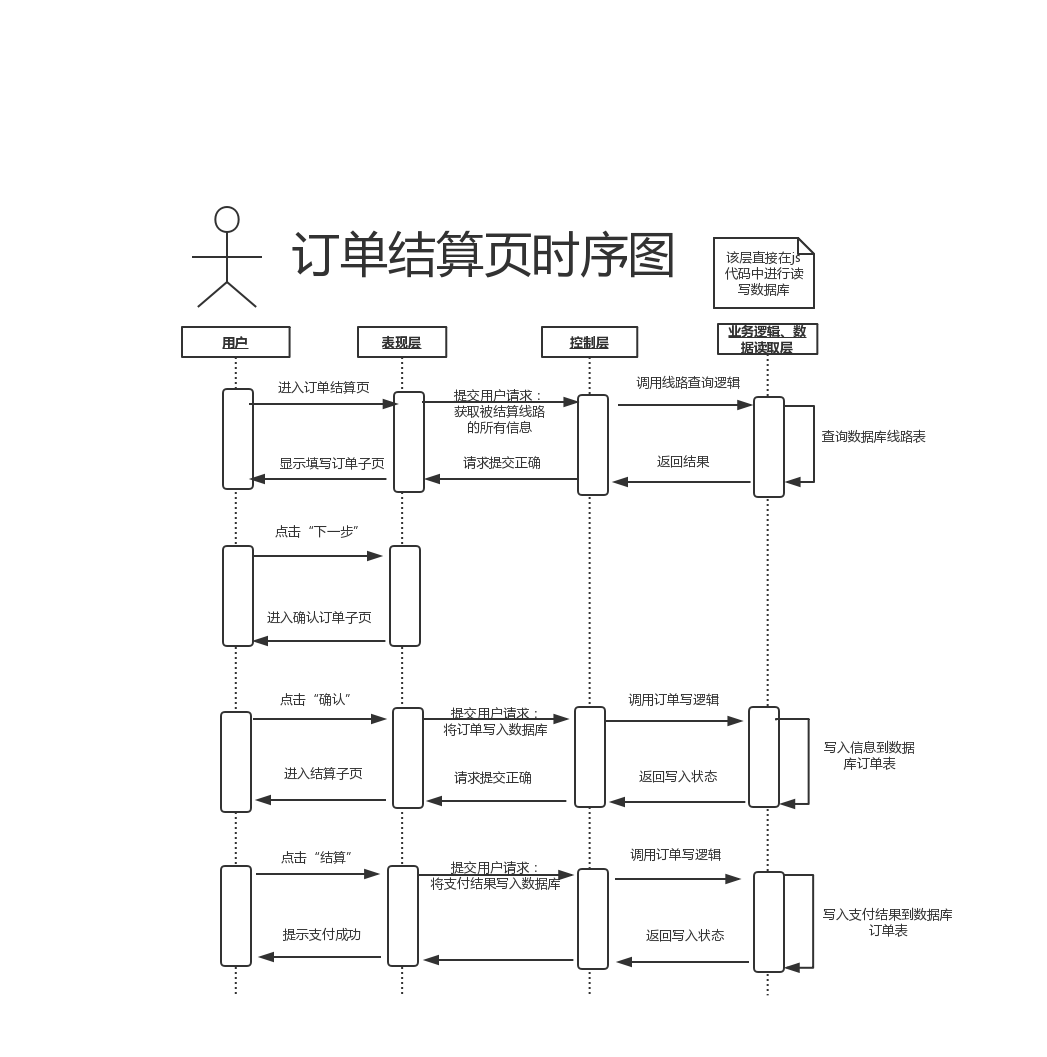
**功能简介：** 用户针对订单的支付功能，用户可以通过路线详情的结账按钮跳转的填写订单界面，确认提交后支付订单。也可以从购物车当中的商品路线状态更改，进入填写订单，确认提交后支付。支付页面跳转会连接到用户的可支付平台，可以链接到支付宝平台，微信平台或者是网上银行卡平台。用户们根据自己常用平台进行订单支付。

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图 4‑15所示。

图 4‑15支付订单操作流程



## 后台管理模块设计

### 模块AM1：管理员登录

**编号：**AM1；

**模块名称：**管理员登录；

**功能简介：**管理员登录可以从该界面进行管理员登录认证，并赋予权限。管理员登录虽然和用户登录是同一个窗口，但是将数据上传到服务器后进行权限认证。反馈的是管理员界面而非用户界面。

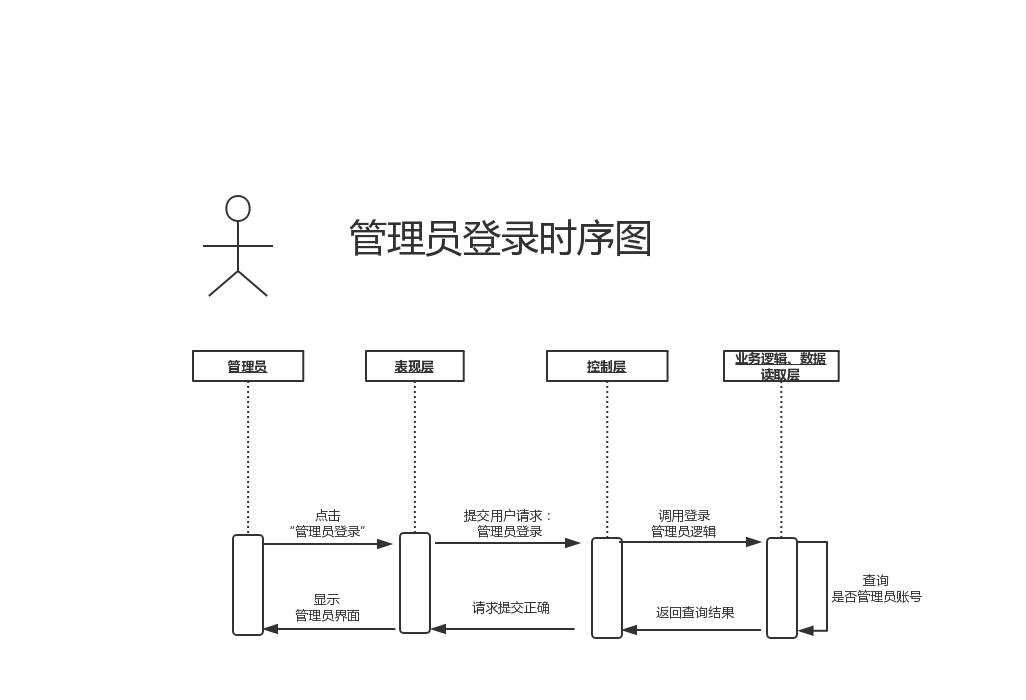
管理员登录后可进行后续的路线信息，路线类型，团购信息修改操作，并将以上操作上传至服务器，服务器再更迭数据反馈到用户界面。

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图 4‑16所示。

图 4‑16管理员登录操作流程



### 模块AM2：路线类型管理

**编号：**AM2；

**模块名称：**路线类型管理；

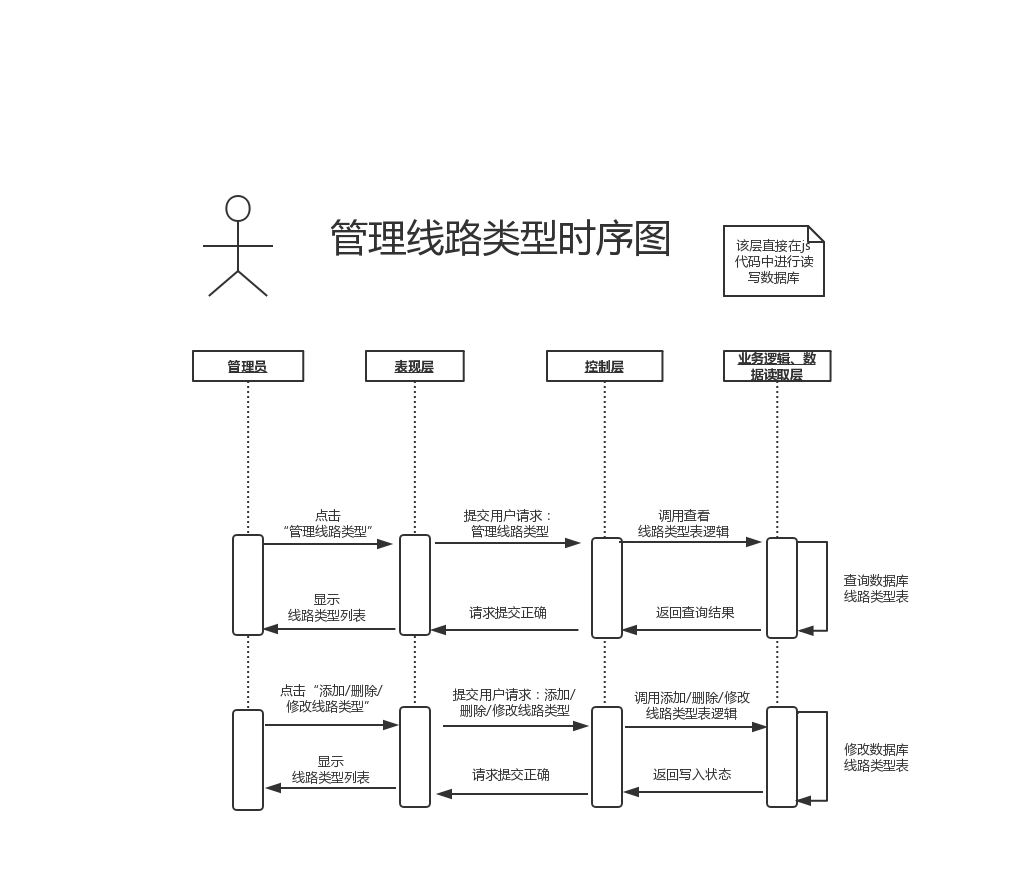
**功能简介：**管理员登录过后，后台数据库为其确认身份并赋予逻辑权限，管理员进入路线类型管理界面，可以针对路线类型的增添，删除，变更。每条路线被修改的信息写入数据库路线表当中，然后链接到路线类型表中。更改后的路线类型会通过后台数据库将变更后的数据反馈到用户界面，然后使得用户界面或购物车存在商品信息进行更改。；

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图 4‑17所示。

图 4‑17 管理路线类型操作流程



### 模块AM3：旅游路线管理

**编号：**AM3；

**模块名称：**旅游路线管理；

**功能简介：**管理员登录过后，后台数据库为其确认身份并赋予逻辑权限，管理员进入路线管理界面，可以针对路线进行增添，删除，变更。每条路线被修改的信息写入数据库路线表当中。或者是对路线的图片进行替换或增添，删除，修改信息会传入后台数据库的线路图片信息表当中。

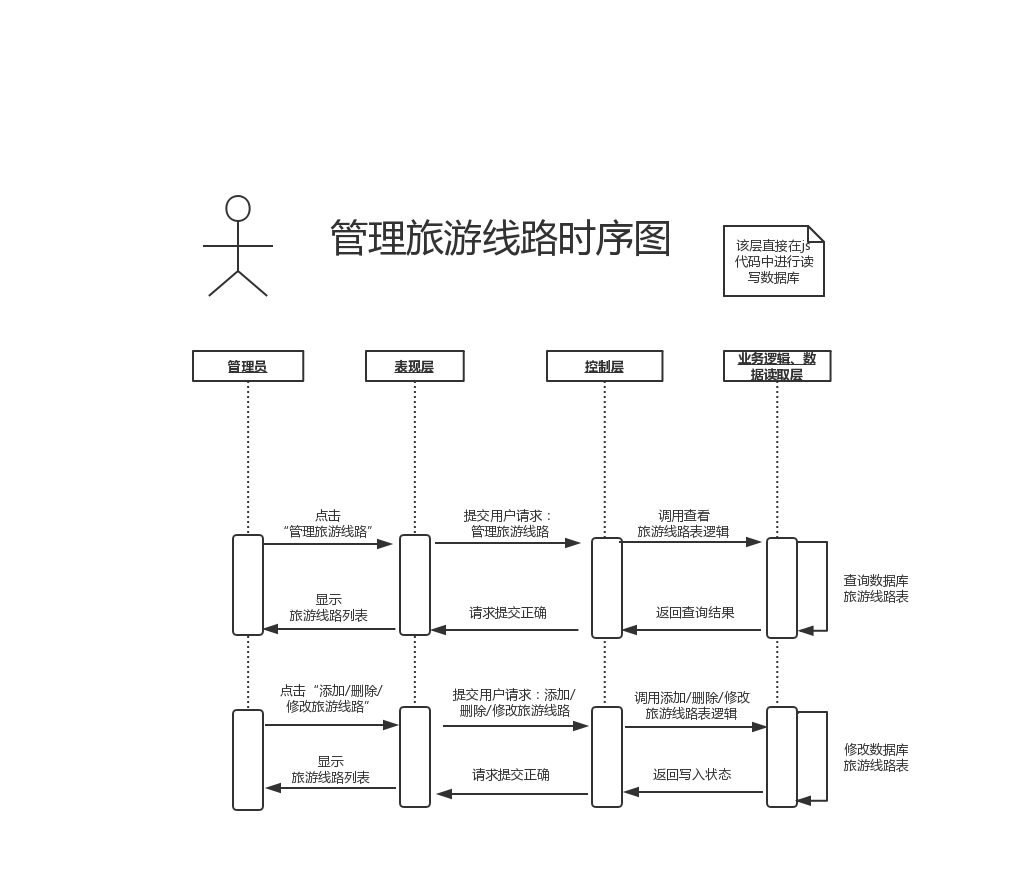
更改后的路线类型会通过后台数据库将变更后的数据反馈到用户界面，然后使得用户界面或购物车存在商品信息进行更改。

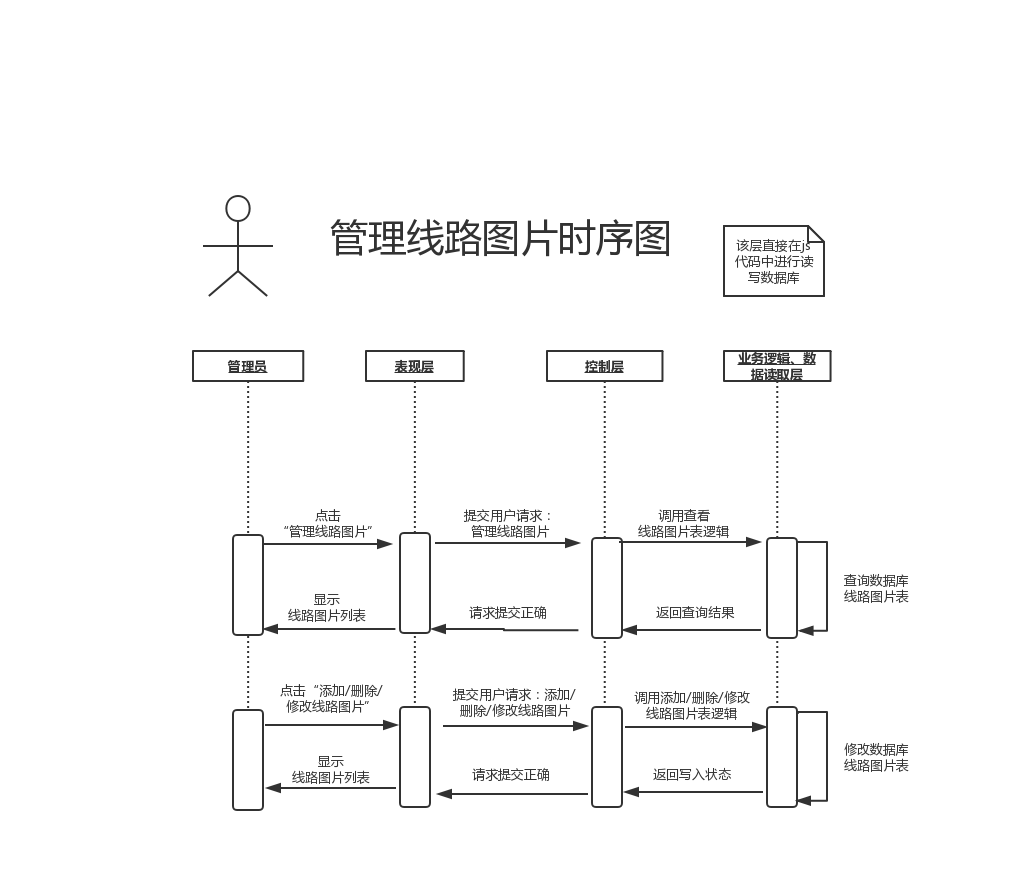
**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图 4‑18所示。

图 4‑18管理路线操作流程





### 模块AM4：团购信息管理

**编号：**AM4；

**模块名称：**团购信息管理；

**功能简介：**管理员登录过后，后台数据库为其确认身份并赋予逻辑权限，管理员进入团购信息管理界面，可以针对团过信息进行增添，删除，变更。每条路线被修改的信息写入数据库路线表团购属性值当中。并且修改后的团购信息会显示在用户浏览界面的团购信息位置处，不会单独在路线详情处显示。

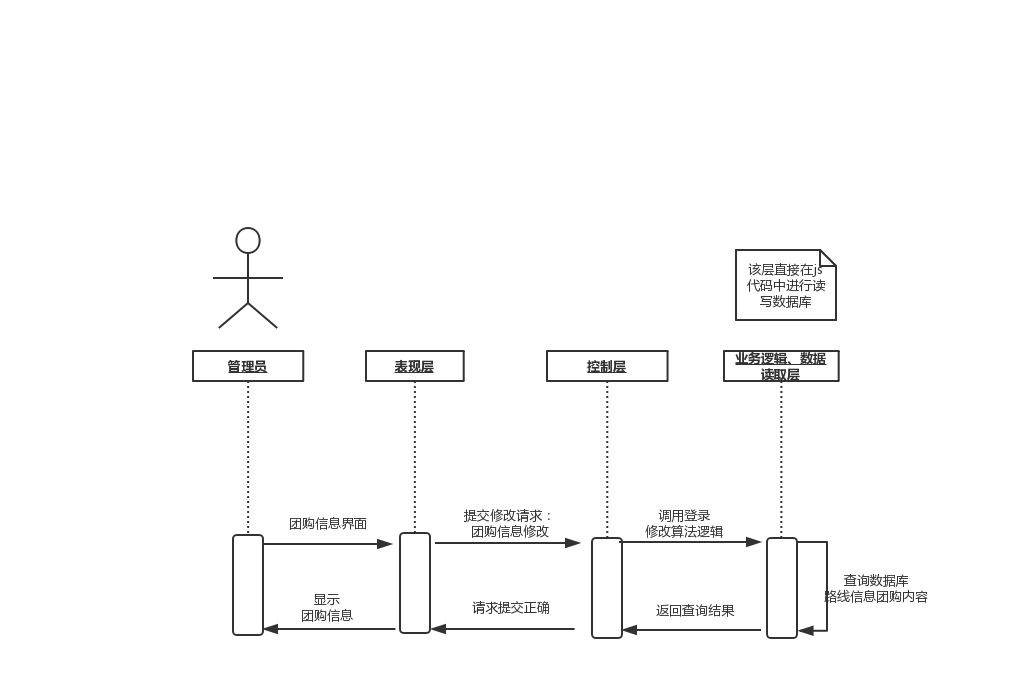
团购信息的变更可以示以提示方式通知关注相应路线的用户，提醒用户该路线开放团购，团购信息删除时也会将出行的所有内容变更，通知相关用户后也会在购物车中存在的相关信息进行变更。但是订单存在的信息不回变更，历史订单内容也不会变更。只是从历史订单中重新访问变更过后的团购路线会跳转到不存在的404错误页面；

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图 4‑19所示。

图 4‑19团购信息管理操作流程



# 数据库设计

## 数据库种类特点

本项目采用了MySQL 5.7.23的数据库。MySQL是一个关系型数据库管理系统，由瑞典MySQL AB公司开发，目前属于Oracle公司，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL具有以下几个特性：

（1）MySQL为多种编程语言提供了API。这些编程语言包括C、C++、Python、Java、Perl、PHP、Eiffel、Ruby,.NET和Tcl等。

（2）优化的 SQL查询算法，有效地提高查询速度。

（3）提供多语言支持，常见的编码如中文的GB2312、BIG5，日文的 Shift\_JIS

等都可以用作数据表名和数据列名。

（4）提供 TCP/IP、ODBC和JDBC等多种数据库连接途径。

（5）提供用于管理、检查、优化数据库操作的管理工具。

（6）支持大型的数据库。可以处理拥有上千万条记录的大型数据库。

（7）MySQL使用标准的SQL数据语言形式。

## 数据库逻辑结构

经过充分的调研和分析，我们将“BrandiTravel平台”的数据中设计了9个实体，其中包括Customer表，LineType表，Line表，Car表，Picture表，OrderDetail表，Tourist表，OT\_Detail表，Manager表。针对每个实体，又有其对应的属性。他们之间的关系主要包括以下几项：

* 客户表，购物车表，路线表，订单表，是独立存在的，用户可以以拥有多个购物车物品，可以查看多条路线及其详情，可以对多个订单执行支付或删除操作，因此用户与购物车，路线，订单是一对多的关系。
* 用户可以加入团购路线，进行组团旅行，而一个团购路线也可以由多个用户参加，所以用户和团购是多对多的关系。
* 线路表和线路存储表于存在一对多的关系，对于一条旅游线路来说，需要至少一张图片进行场景的描述，多张图片可以更多地使得客户们对旅游路线的深层了解。
* 用户和出游游客信息是一对多的关系，用户可以填写多名出游人员信息，但是每名游客信息与订单游客信息是一对一的关系。

综上分析，BrandiTravel平台前端的实体关系图（概念数据模型）如下图 5‑1所示。

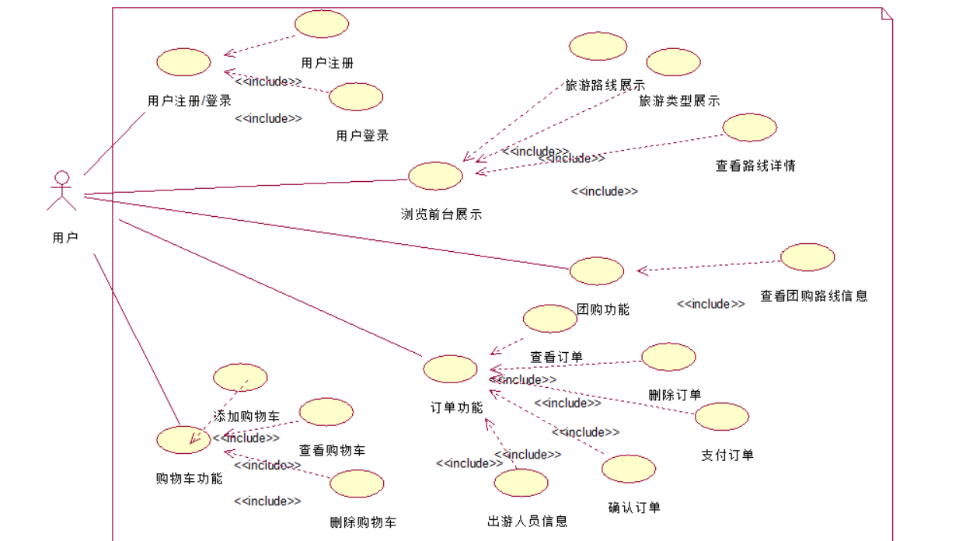


图 5‑1 前端实体关系图

## 物理结构设计

根据上面的实体关系，设计数据库表以及根据数据模型图 5‑2可知，系统一共应有9张表。下面是关于表的详细说明。

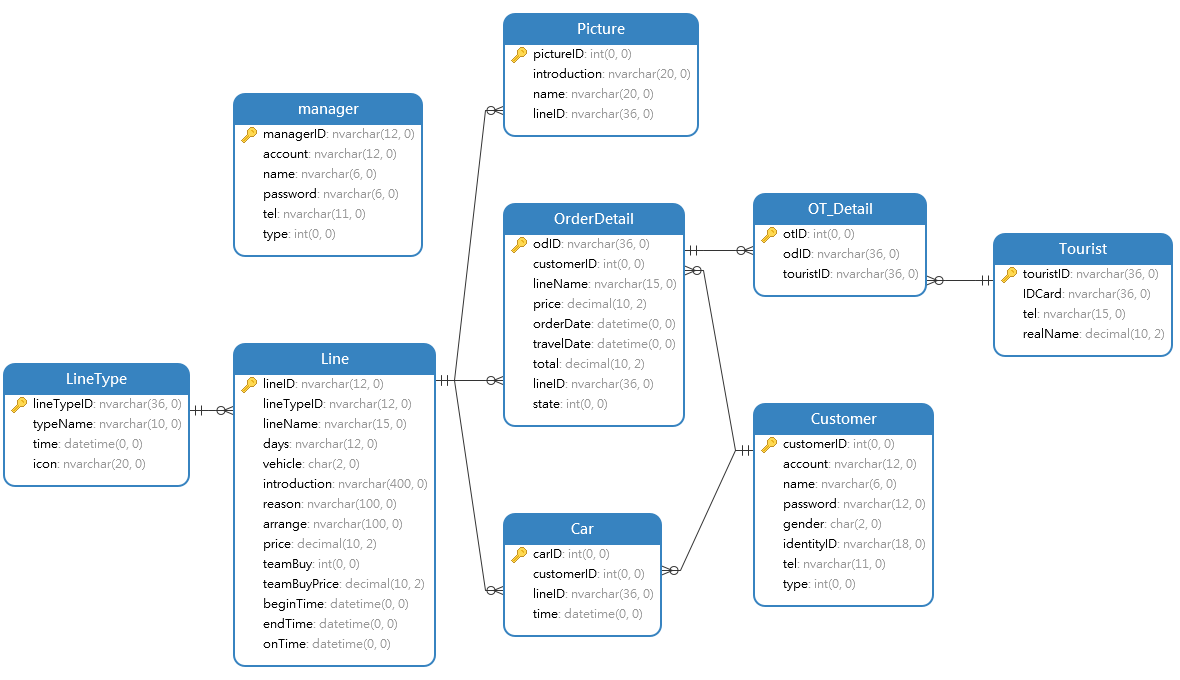


图 5‑2 数据库表的关系图

### 表1：Customer表

用户表标识用户信息，其表结构如表 5‑1下：

表 5‑1 客户表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 允许为空 | 备注 |
| customerID | int | 否 | 客户编号，主键，自增长 |
| account | nvarchar(12) | 否 | 客户账号，唯一键 |
| name | nvarchar(6) | 否 | 客户真实姓名 |
| password | nvarchar(12) | 否 | 密码 |
| gender | char(2) | 否 | 性别 |
| identityID | nvarchar(18) | 否 | 客户身份证号 |
| tel | nvarchar(11) | 否 | 客户手机号 |
| type | int | 是 | 状态，值为1时表示为系统 |

### 表2：LineType表

旅游路线类型表的是旅游路线类型的信息，其表结构如表 5‑2下：

表 5‑2 旅游路线类型表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 允许为空 | 备注 |
| lineTypeID | nvarchar(36) | 否 | 线路类型编号，主键 |
| typeName | nvarchar(10) | 否 | 线路类型名称 |
| time | datetime | 否 | 线路类型添加时间 |
| icon | nvarchar(20) | 否 | 线路类型图标名 |

### 表3：Line表

路线详情表标识的是该路线存在的详细介绍，其表结构如表 5‑3下：

表 5‑3 路线详情表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 允许为空 | 备注 |
| lineID | nvarchar(12) | 否 | 线路编号，主键 |
| lineTypeID | nvarchar(12) | 否 | 线路类型编号，外键 |
| lineName | nvarchar(15) | 否 | 线路名称 |
| days | nvarchar(12) | 否 | 出游天数 |
| vehicle | char(2) | 否 | 出游交通工具 |
| introduction | nvarchar(400) | 是 | 线路简介 |
| reason | nvarchar(100) | 是 | 线路推荐理由 |
| arrange | nvarchar(100) | 是 | 线路行程安排 |
| price | decimal(10, 2) | 否 | 线路价格 |
| teamBuy | int | 是 | 是否为团购线路，值为1时表示为团购线路 |
| teamBuyPrice | decimal(10, 2) | 是 | 线路团购价格 |
| beginTime | datetime | 是 | 线路团购开始时间 |
| endTime | datetime | 是 | 线路团购结束时间 |
| onTime | datetime | 否 | 线路添加时间 |
|  |  |  |  |

### 表4：Car表

购物车表标识的是用户对自己购物管理的一个功能表，其表结构如表 5‑4下：

表 5‑4 购物车表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 允许为空 | 备注 |
| carID | int | 否 | 购物车编号，主键，自增长 |
| customerID | int | 否 | 客户编号，外键 |
| lineID | nvarchar(36) | 否 | 线路编号，外键 |
| time | datetime | 否 | 线路添加至购物车的时间 |

### 表5：Picture表

线路图片表是每条线路存储其详情的后台表，其表结构如表 5‑5下：

表 5‑5 线路图片表

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| 字段 | 类型 | 允许为空 | 备注 |
| pictureID | int | 否 | 图片编号，主键，自增张 |
| introduction | nvarchar(20) | 否 | 图片介绍 |
| name | nvarchar(20) | 否 | 图片名称 |
| lineID | nvarchar(36) | 否 | 线路编号，外键 |

### 表6：OrderDetail表

订单详情表是用户查看订单内容所需要的表单，可根据详情进行修。其表结构如表 5‑6下：

表 5‑6 订单详情表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| 字段 | 类型 | 允许为空 | 备注 |
| odID | nvarchar(36) | 否 | 订单编号，主键 |
| customerID | int | 否 | 客户编号，外键 |
| lineName | nvarchar(15) | 否 | 线路名称 |
| price | decimal(10, 2) | 否 | 线路价格 |
| orderDate | datetime | 否 | 下订单日期 |
| travelDate | datetime | 否 | 出游日期 |
| total | decimal(10, 2) | 否 | 总价 |
| lineID | nvarchar(36) | 否 | 线路编号，外键 |
| state | int | 否 | 订单显示状态，0为不显示，1为显示 |

### 表7：Tourist表

游客信息表是出行游客自身出游的信息，其表结构如表 5‑7下：

表 5‑7 游客信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| 字段 | 类型 | 允许为空 | 备注 |
| touristID | nvarchar(36) | 否 | 游客信息编号，主键 |
| IDCard | nvarchar(36) | 否 | 游客身份证号 |
| tel | nvarchar(15) | 否 | 游客手机号 |
| realName | decimal(10, 2) | 否 | 游客真实姓名 |

### 表8：OT\_Detail表

订单游客信息表是订单所含的出游人员信息，这样可以保证旅游路线的游客是一一对应的。其表结构如表 5‑8下：

表 5‑8 订单游客信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| 字段 | 类型 | 允许为空 | 备注 |
| otID | int | 否 | 订单游客信息编号，主键自增长 |
| odID | nvarchar(36) | 否 | 订单编号，外键 |
| touristID | nvarchar(36) | 否 | 游客信息编号，外键 |



### 表8：Manager表

管理员信息表是订单所含的后台系统管理员信息，这样管理员登录时可以由后台数据库查询管理员信息是否对等，然后在提交正确查询结果后赋予权限登录管理员后台系统管理页面。其表结构如表 5‑9下：

表 5‑9 管理员信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| 字段 | 类型 | 允许为空 | 备注 |
| ManagerID | int | 否 | 管理员编号，主键，自增长 |
| account | nvarchar(12) | 否 | 管理员账号，唯一键 |
| name | nvarchar(6) | 否 | 管理员真实姓名 |
| password | nvarchar(12) | 否 | 密码 |
| tel | nvarchar(11) | 否 | 管理员手机号 |
| type | int | 是 | 状态，值为1时表示为系统 |

# 界面设计

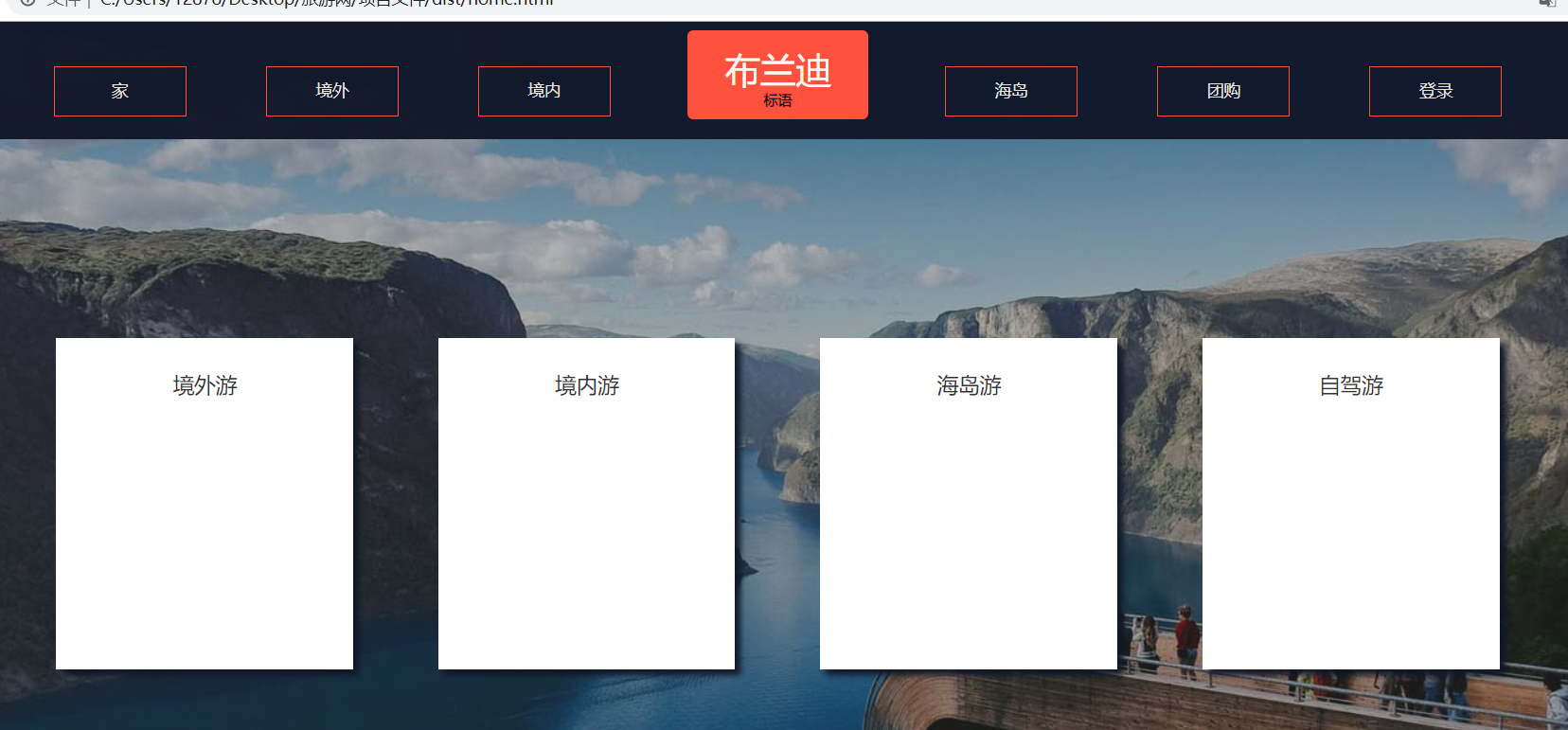
## 首页设计

BrandiTravel的首页是按照图6-1的界面设计的。网页的顶部是网站的主要功能，包含了境内游，境外游，海岛游，团购功能选择。左边的“家”则是首页的跳转按钮。右上角为用户登录界面。以上用户们可以根据自己的意愿进行功能选择。

网页的主要部分则是旅游景点的图片轮转图。从各地挑选的美景为其搭配。轮转图则由后天数据库管理员进行图片更新。当然点击图片也可以进行相应的旅游景点的跳转。

左下角的购物车按钮，可以直接从主页面跳转至购物车界面，可进行当路线选择后进行购物车查看，或移除，或结账功能，由用户们自行决定，如图 6‑1所示。

图 6‑1 首页设计图

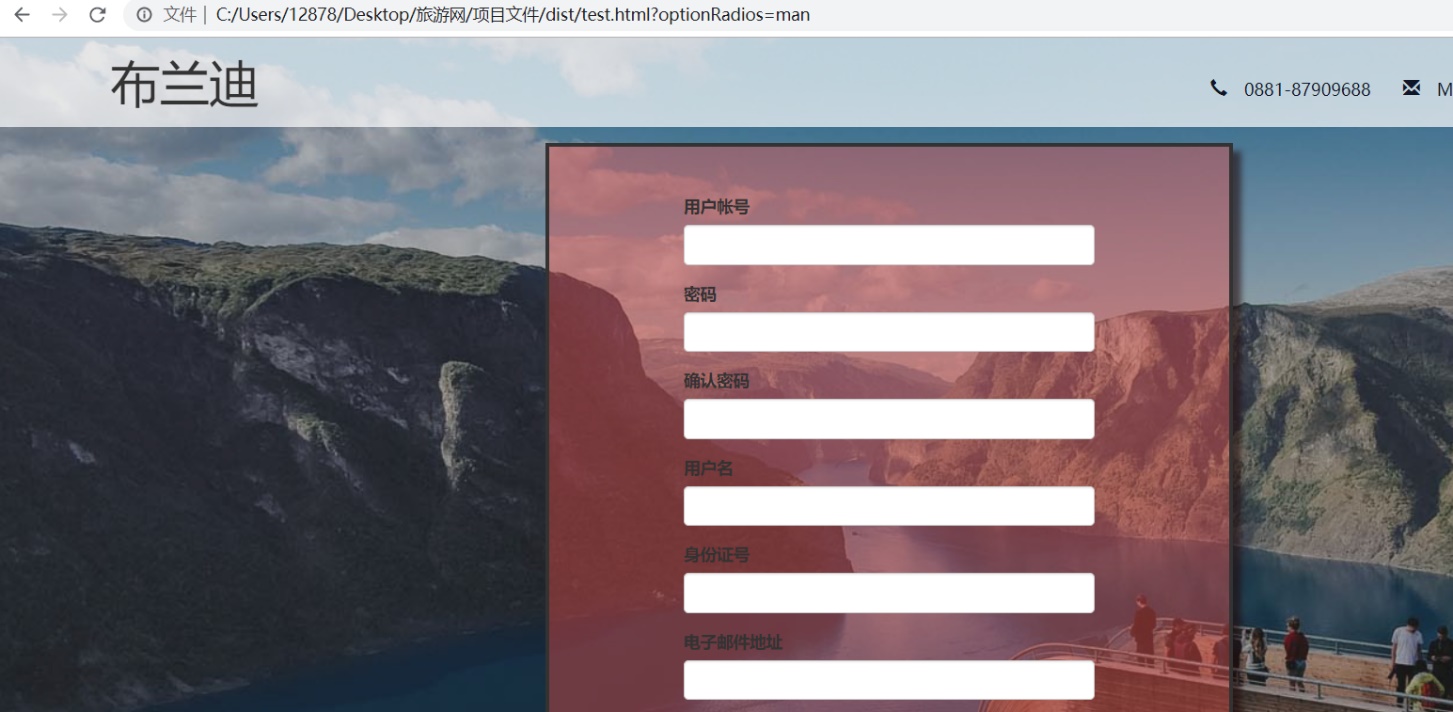


## 用户注册/登录页面设计

用户注册/登录给网站提供用户们对网站的访问权限。用户们通过登录获得网站的访问与功能实现权限。

用户的注册需要填写用户账号，密码，用户名，电子邮件，这些数据会上传至后台数据库，然后对应由数据库管理员管理。然后填写身份证号实名制注册，保证每个用户的真实存在，确保了网络黄牛的存在会被管理并遏制。用户注册/登录界面如图 6‑2，图 6‑3所示。

图 6‑2 用户注册页面



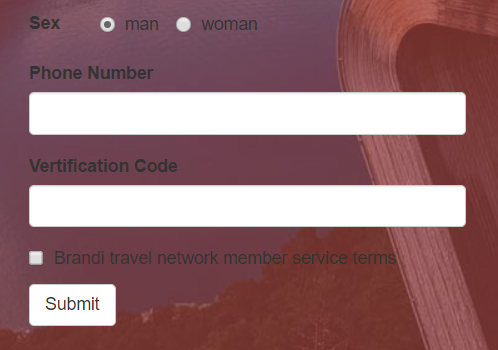


图 6‑3 用户登录页面图



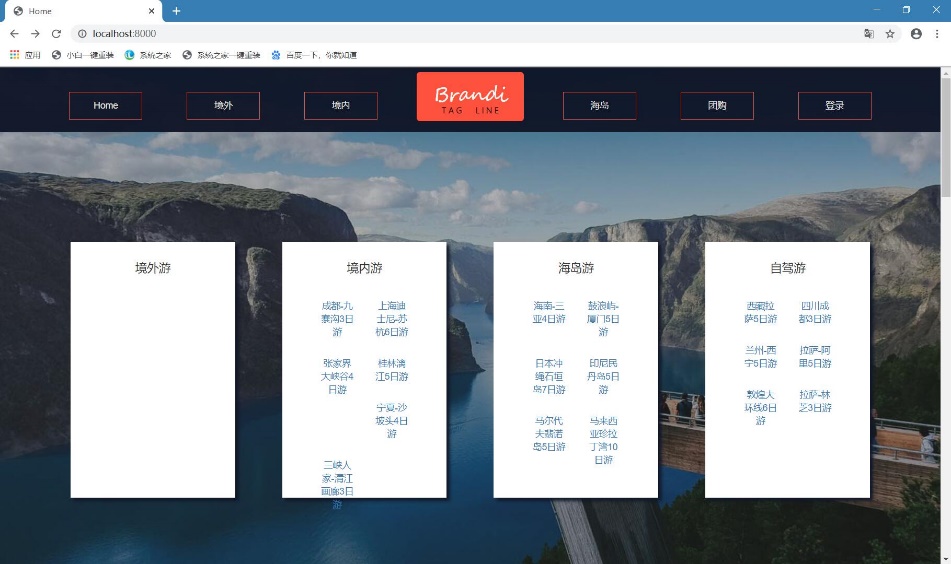
## BrandiTravel旅游页面设计

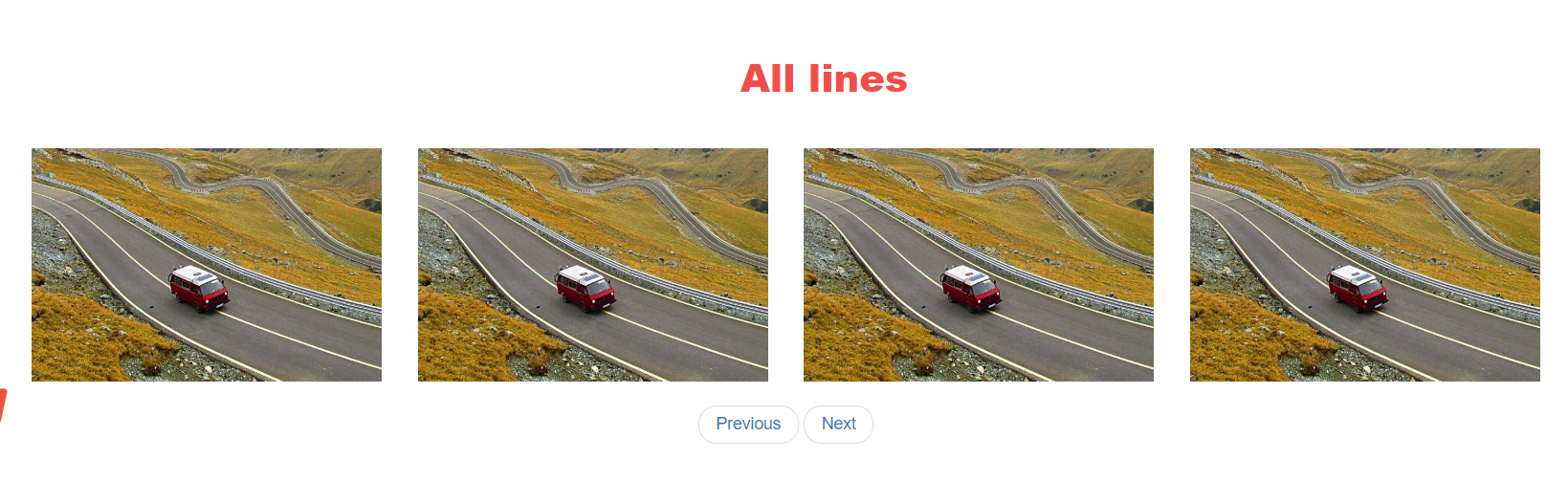
### 旅游路线概况页面设计

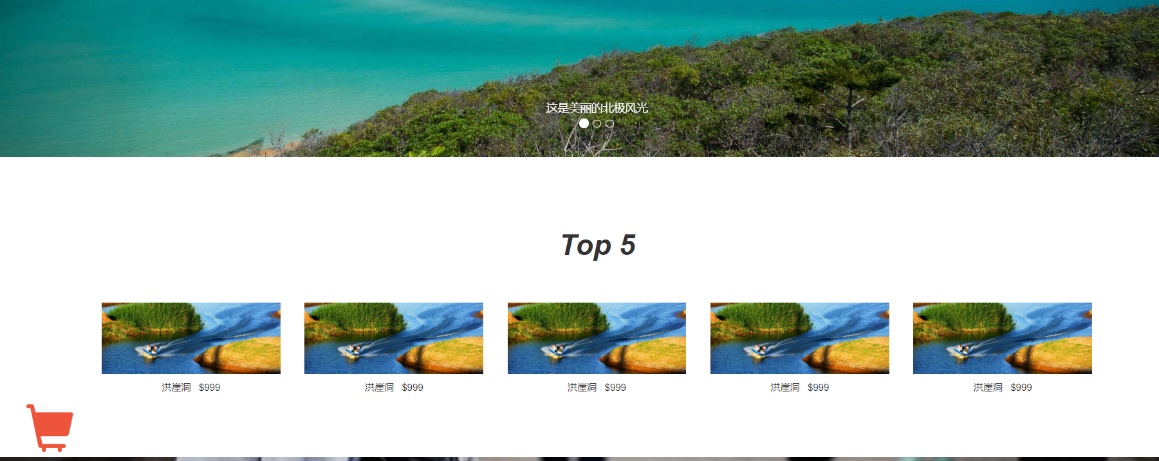
用户们可以从该界面直接对网页路线进行大致浏览，由后台数据库连接的路线景点图片，以及简要的基础介绍。所以当用户们对其产生兴趣后可直接点击路线，进入路线详情进行更多细节浏览。

路线浏览可以从主页面进行下拉滑动条浏览，也可以进行路线类型进行分类浏览，团购路线中可查看可团购路线浏览。其中路线浏览也有“前五名旅游路线推荐”，通过对浏览人气，景点评价和订购数量分析而得的前五名人气景点，如图 6‑4所示。

图 6‑4 BrandiTravel路线概况页面设计图







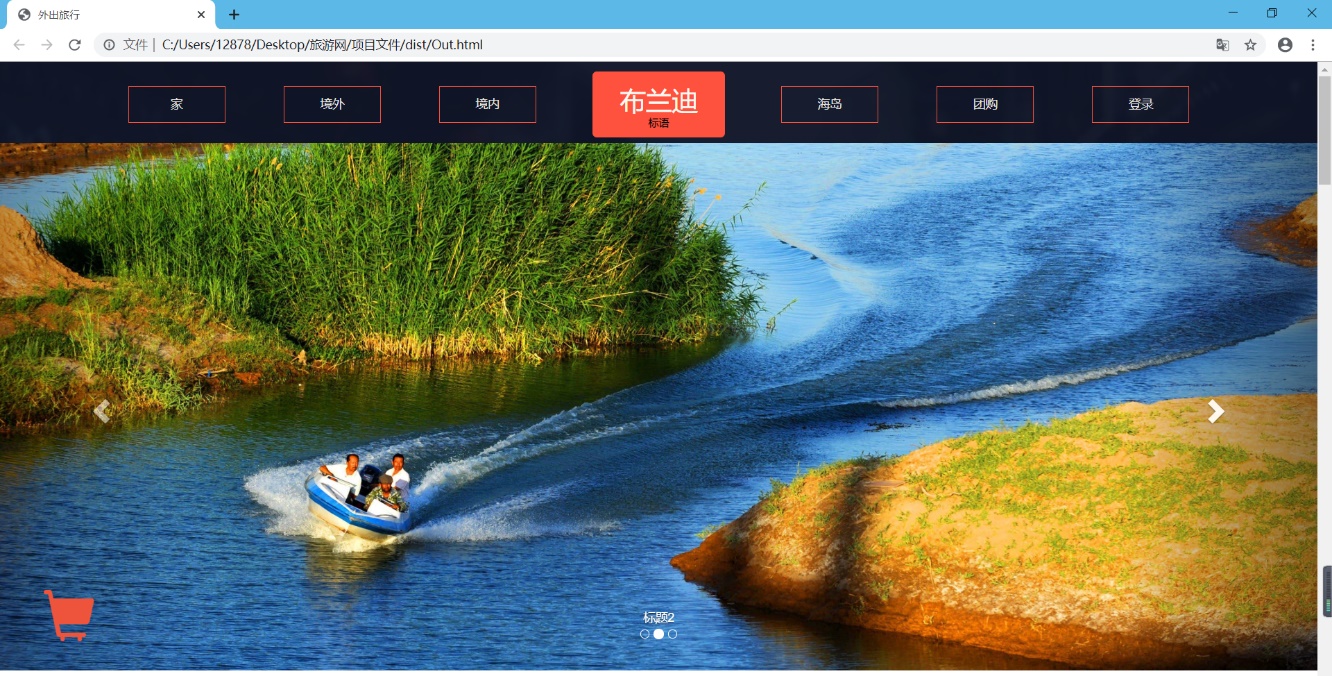
### 旅游路线类型页面设计

用户可以进入旅游路线类型界面进行不同路线分类，选择自己想去的旅游路线方案。现阶段主要分类为境外游，境内游，海岛游，自驾游三个方面，以后可以进行多种方案进行增添。

路线的分类由后台管理员进行处理分类，其中管理员为路线进行概况描述后，再对路线进行系统分类。

用户可以进入不同类型的旅游线路，然后对其感兴趣的路线在进入详情介绍，如图 6‑5所示。

图 6‑5 BrandiTravel路线类型页面设计图





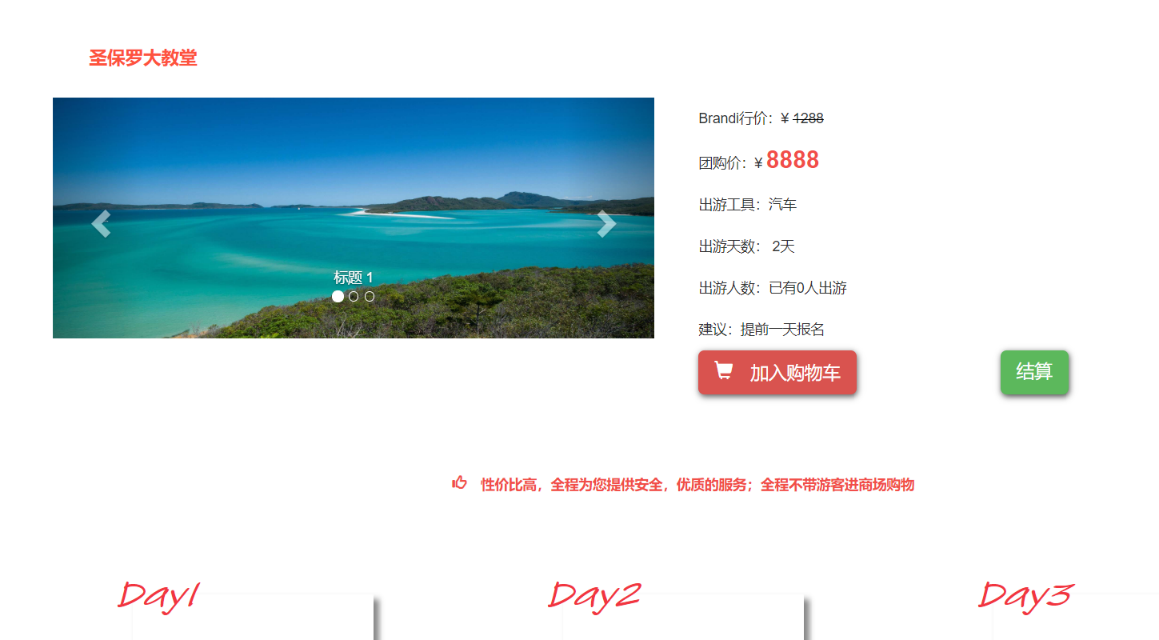
### 旅游路线详情页面设计

用户若对某一旅游路线产生兴趣，可以点击该路线，进入路线详情对路线景点的简介，景点美图，原价格，BrandiTravel折扣价，出游路线天数，出游路线工具，出游人数进行浏览。

当用户对该路线有想法的时候，可以将其加入购物车以方便考虑支付或移除操作。当用户直接心动了，就可以立即对该路线进行支付操作，进入支付订单界面为该路线结账。

用户如果想对该路线进行其他了解，可以拨打客服热线进行更多细节了解。也可以浏览评论区（功能暂未开放），浏览参考其他用户对该景点的介绍与路线经历分享介绍，如图 6‑6所示。

图 6‑6 BrandiTravel路线详情页面设计图



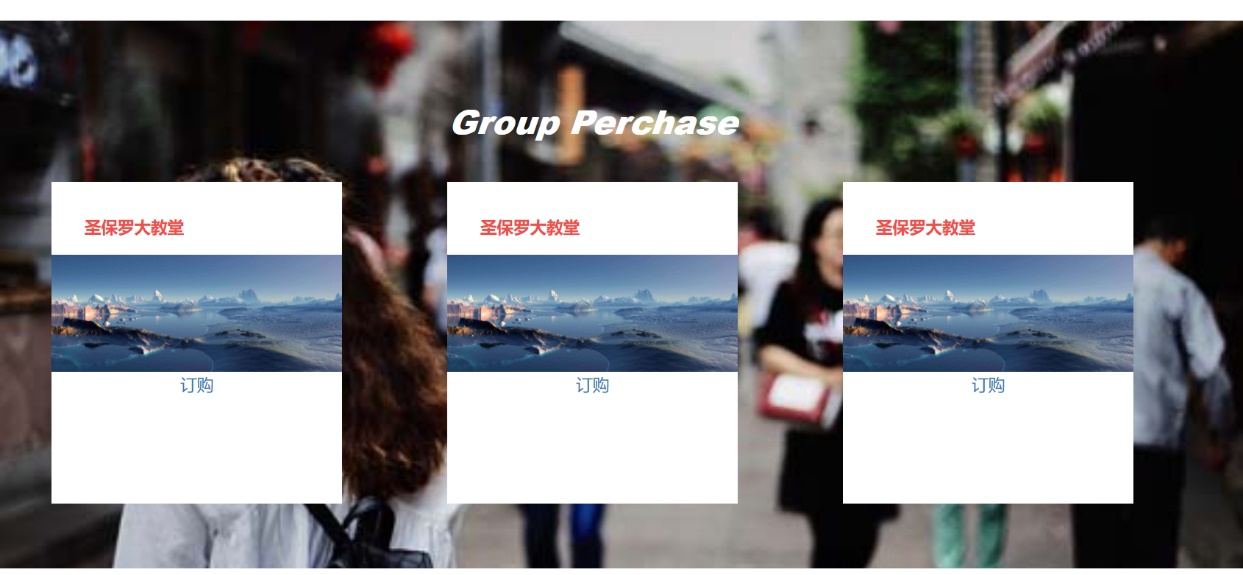
### 团购旅游路线页面设计

用户如果对某一条线路产生浓厚的兴趣，但是对于出游人数过多，或者是对于人数过少，不愿意在出行费用上过多开销。可以考虑点击团购的旅游路线，进入该界面进行团购旅游路线了解。

当然也会团购路线不会针对每一条旅游路线，可能会对人气爆棚的路线，为了人流量考虑而设置团购旅游路线，也有可能是某一条路线暂时人数稀少时，正好可以让大家领略自然风光时添加入团购路线当中。

所以团购旅游路线是针对景点自身进行开放，存在用户想要的旅游路线景点未被添加到团购旅游路线当中，可能会造成部分用户的遗憾，但这也是无法避免的原因之一，如图 6‑7所示。

图 6‑7 BrandiTravel团购页面设计图



## BrandiTravel购物车页面设计

用户们对中意的旅游路线可以在其详情中点击“加入购物车”按钮，使其添加到购物车当中。然后用户们可以通过主页上的购物车按钮进入购物车中查看。

查看购物车直接可以进入购物车界面查看。购物车存在的功能包括购物车路线商品序号，购物车路线商品图片简介，购物车路线商品名称，购物车路线商品价格，购物车路线商品订单状态，购物车路线商品操作。

若用户们对某一商品存在些许淡忘，可以点击路线商品图片或名称进入路线详情进行查看。

若用户们想对路线商品进行操作更改，可以点击操作中的移除，这样就可以将购物车中的不想要的商品移除购物车，当然添加购物车只能在相关中意的路线详情中添加。

对于下单状态的商品，可以在订单详情进行其他操作。购物车可在未来开发清空功能（暂未开放），可以一键删除或者是一次性全结账，如图 6‑8所示。

图 6‑8 BrandiTravel购物车页面设计图



## BrandiTravel订单页面设计

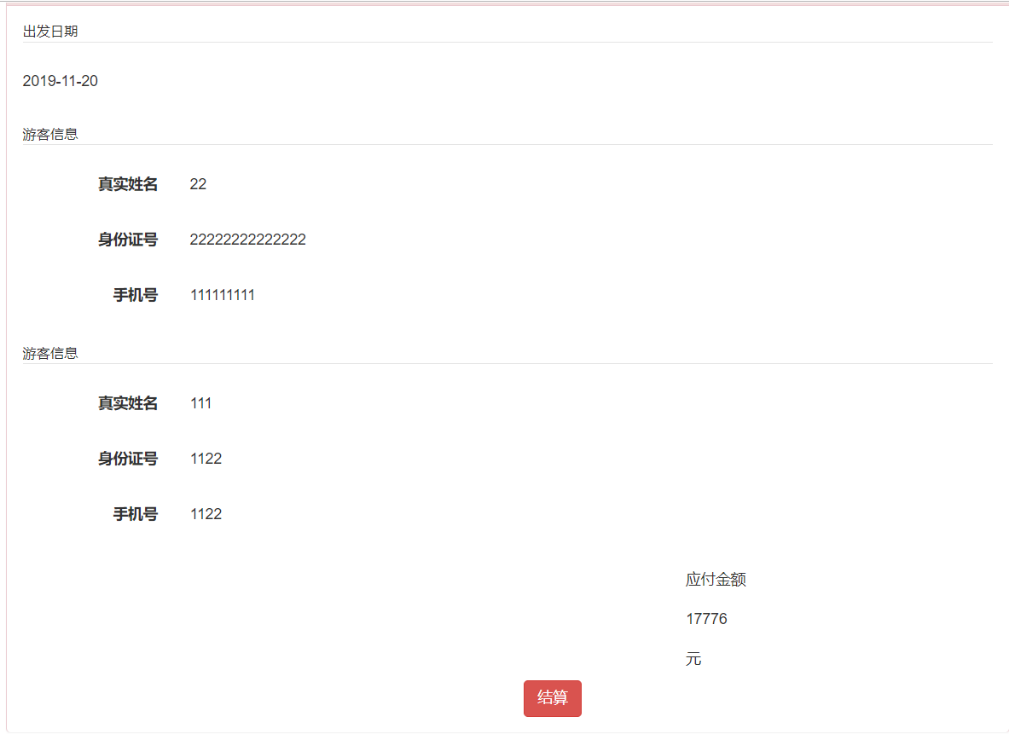
### 订单界面

用户进入订单界面可以进行相关订单操作。其中包括对订单详情查看，对订单的支付操作，订单的删除操作。

进入订单界面可以看见首先是填写订单，其中包括该订单的线路编号，线路名称。线路单价，线路出行工具。填写完毕后可进入确认订单信息，最后提交并支付订单。

对于已支付过后的订单然后可以进行再次查看其详情并且可以对该订单进行删除或者详情浏览操作，如图 6‑9所示。

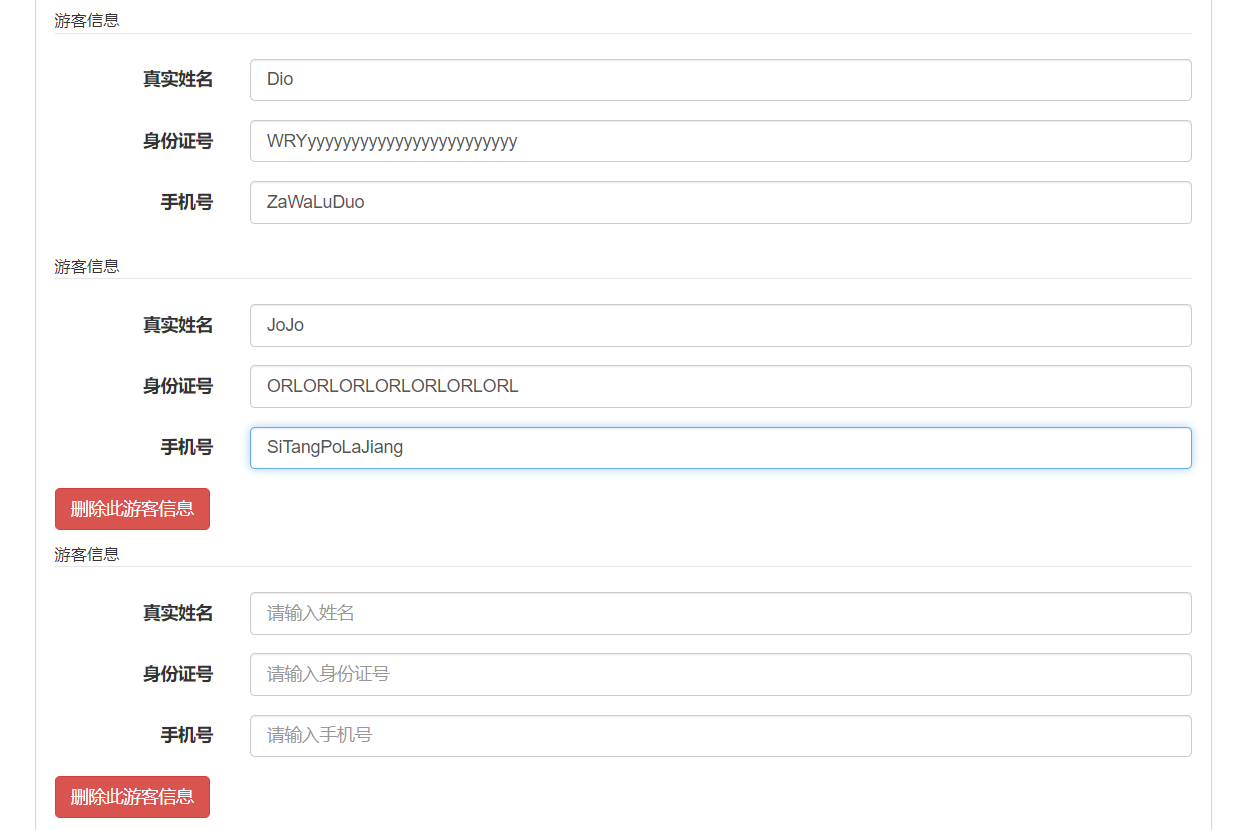
图 6‑9 BrandiTravel订单页面设计图



### 填写出游人员信息页面

用户在填写订单时，可以将出行的人员的信息填写入表，然后在提交订单后把这些数据输入到后台服务器。包括了游客的真实姓名，身份证号，手机号主要的个人信息。如图 6‑10所示

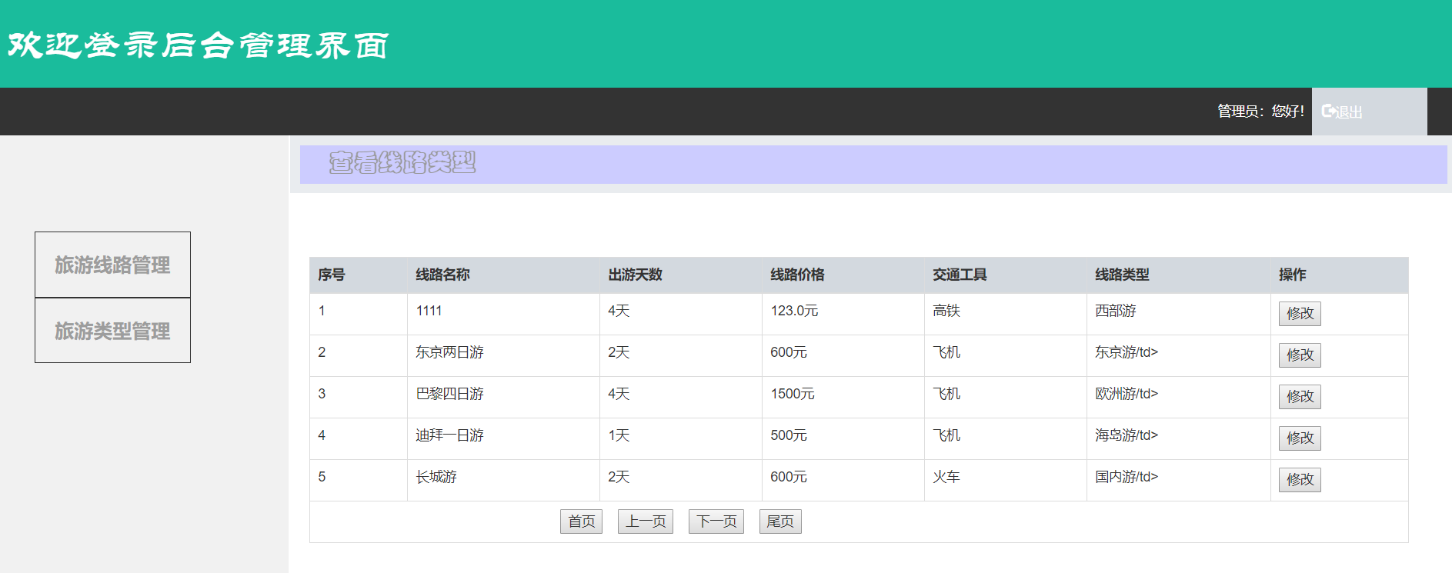
图 6‑10 BrandiTravel填写出游人员信息页面

## 后台主页面设计

后台主界面显示后台基本信息，其中包括了路线管理和路线类型管理功能。管理员的退出窗口在右上角选择账户退出，退出后操作保留于数据库当中。具体设计如图 6‑12所示。

图 6‑12 后台主页面设计图



## 后台管理员登录页面设计

管理员登录对管理员提供对登录网站的服务，登录界面虽然与用户共用同一界面，但是相应的账号传入会由服务器反馈显示不同的界面。登录界面如图 6‑11所示。

图 6‑11后台管理员登录页面设计图



## 旅游路线管理页面设计

管理员登录后，可以直接对路线的进行查看，页面首页默认界面时路线信息查看，包括了路线的名称，出游天数，出游价格，出行工具，线路类型。每一条线路后的“修改”按钮可以直接对改线路进行属性修改。比如，可以将东京游改为5天的出游天数，改为汽车的出行方式。每一次修改都会在管理退出后保存在后台数据库当中，然后用户在刷新界面后出现修改内容更改。

若当出现用户正在浏览相应的路线，且管理员也正在对该路线进行修改，这可能会造成用户当时的下单造成价格或路线属性不一的情况。所以管理员退出后每一次整体数据更新不是及时更新，而是后台数据在定点时间进行网页数据更新，这样用户也能把握准确时间再进行浏览和决定。如图 6‑13所示

图 6‑13 旅游路线管理页面设计图



## 旅游路线类型管理页面设计

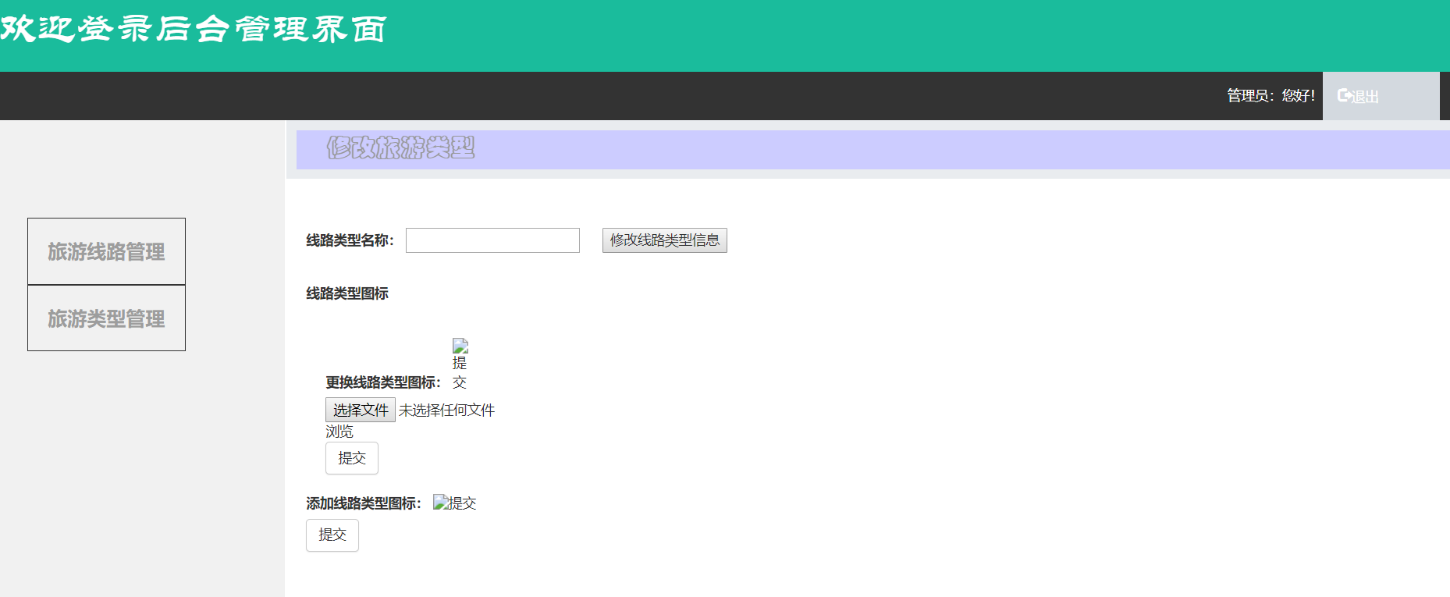
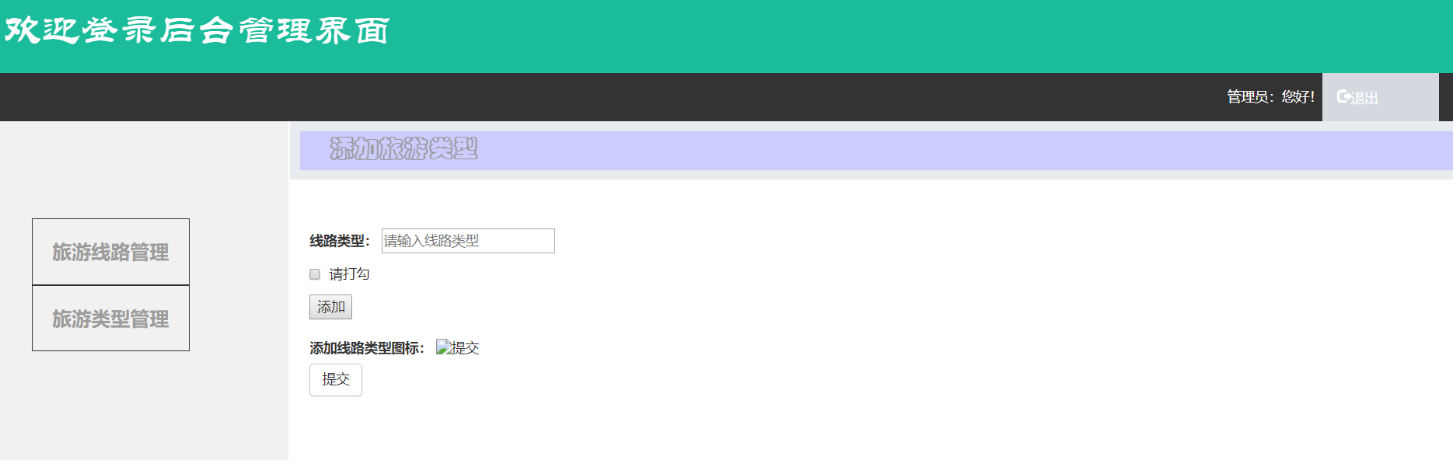


管理员登录后，管理员可以通过进入旅游路线管理页面对路线类型进行修改。首先是点击左侧的旅游类型管理按钮进入查看旅游路线类型页面，进入页面后的请求由网页层层调用，直到数据库返回请求，将路线类型表数据返回到页面。

然后再进行其他路线类型管理操作，在路线类型展示界面包括了路线名称，路线类型的编号，可以更换为相应的类型。然后是后面的类型状态修改功能，可以直接进行修改或者删除。在路线类型管理按钮也有下拉菜单选项，包含了修改和删除功能。

若当出现用户正在浏览相应的路线类型，且管理员也正在对该路线类型进行修改，这可能会造成用户当时的下单造成价格或路线属性不一的情况。所以管理员退出后每一次整体数据更新不是及时更新，而是后台数据在定点时间进行网页数据更新，这样用户也能把握准确时间再进行浏览和决定。如图 6‑14所示

图 6‑14 旅游路线类型管理页面设计图



## 团购旅游路线管理页面设计

暂未开发