**BrandiTravel详细设计**

**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](#_Toc25216263)

[1.1 目的 1](#_Toc25216264)

[1.2 范围 1](#_Toc25216265)

[1.3 引用标准 1](#_Toc25216266)

[1.4 参考资料 2](#_Toc25216267)

[1.5 版本更新信息 2](#_Toc25216268)

[2. 详细设计简述 3](#_Toc25216269)

[2.1 设计简介 3](#_Toc25216271)

[2.2. 模块简介 4](#_Toc25216275)

[3. 前台展示业务模块详细设计 7](#_Toc25216280)

[3.1 用户登录模块 7](#_Toc25216282)

[3.1.1. 表现层 8](#_Toc25216287)

[3.1.2. 控制层 8](#_Toc25216288)

[3.1.3. 业务逻辑数据访问层 9](#_Toc25216289)

[3.2 用户注册模块 9](#_Toc25216298)

[3.2.1. 表现层 10](#_Toc25216311)

[3.2.2. 控制层 11](#_Toc25216315)

[3.2.3. 业务逻辑数据访问层 12](#_Toc25216316)

[3.3 首页旅游路线展示模块 12](#_Toc25216317)

[3.3.1. 表现层 13](#_Toc25216319)

[3.3.2. 控制层 13](#_Toc25216320)

[3.3.3. 业务逻辑数据访问层 14](#_Toc25216321)

[3.4 旅游路线类型展示模块 15](#_Toc25216322)

[3.4.1. 表现层 16](#_Toc25216324)

[3.4.2. 控制层 17](#_Toc25216325)

[3.4.3. 业务逻辑数据访问层 17](#_Toc25216326)

[3.5 旅游路线详情展示模块 18](#_Toc25216327)

[3.5.1. 表现层 19](#_Toc25216329)

[3.5.2. 控制层 20](#_Toc25216330)

[3.5.3. 业务逻辑数据访问层 21](#_Toc25216331)

[3.6 团购路线展示模块 21](#_Toc25216332)

[3.6.1. 表现层 22](#_Toc25216334)

[3.6.2. 控制层 22](#_Toc25216335)

[3.6.3. 业务逻辑数据访问层 23](#_Toc25216336)

[3.7 购物车功能模块 23](#_Toc25216337)

[3.7.1. 表现层 24](#_Toc25216339)

[3.7.2. 控制层 25](#_Toc25216340)

[3.7.3. 业务逻辑数据访问层 26](#_Toc25216341)

[3.8 订单功能模块 27](#_Toc25216342)

[3.8.1. 表现层 28](#_Toc25216362)

[3.8.2. 控制层 29](#_Toc25216363)

[3.8.3. 业务逻辑数据访问层 30](#_Toc25216364)

[4. 后台管理系统的详细设计 31](#_Toc25216365)

[4.1. 后台管理层模块表现层 31](#_Toc25216370)

[4.2. 后台管理模块控制层 32](#_Toc25216371)

[4.3. 后台管理模块业务逻辑数据访问层 33](#_Toc25216372)

[5. 公共部分模块详细设计 35](#_Toc25216373)

[5.1 公共页面 35](#_Toc25216379)

[5.2 业务逻辑数据访问组件 35](#_Toc25216380)

# 导言

## 目的

该文档主要是根据BrandiTravel项目概要设计进行深入层次的详细描述，内容主要对系统的类型，模块，以及功能上的详细设计。其中包括了用户注册，用户登录，旅游产品展示，旅游路线类型，旅游路线管理，购物车，订单等功能模块的详细设计。

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

该文档内容所包含了对BrandiTravel的功能模块，主要模型进行了详细描述以及细节代码展示。

## 引用标准

[1]《SPM 课程网站详细设计》文档 北京邮电大学

## 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 | 初始发布版本 |
| 002 | 2019.11.5 | 2.0 | 局部 | 部分内容修改 |
| 003 | 2019.11.20 | 3.0 | 局部 | 完善修改 |

# 详细设计简述



## 设计简介

BrandiTravel网站项目在设计思路上采用 MVC 的设计模式---Model 模型层、View 视图层、Controller 控制层；在体系架构上选择用 Rest API 架构，形成如下的三层结构：

表现层、控制层、业务逻辑数据访问层、数据访问层如图 2‑1 所示。

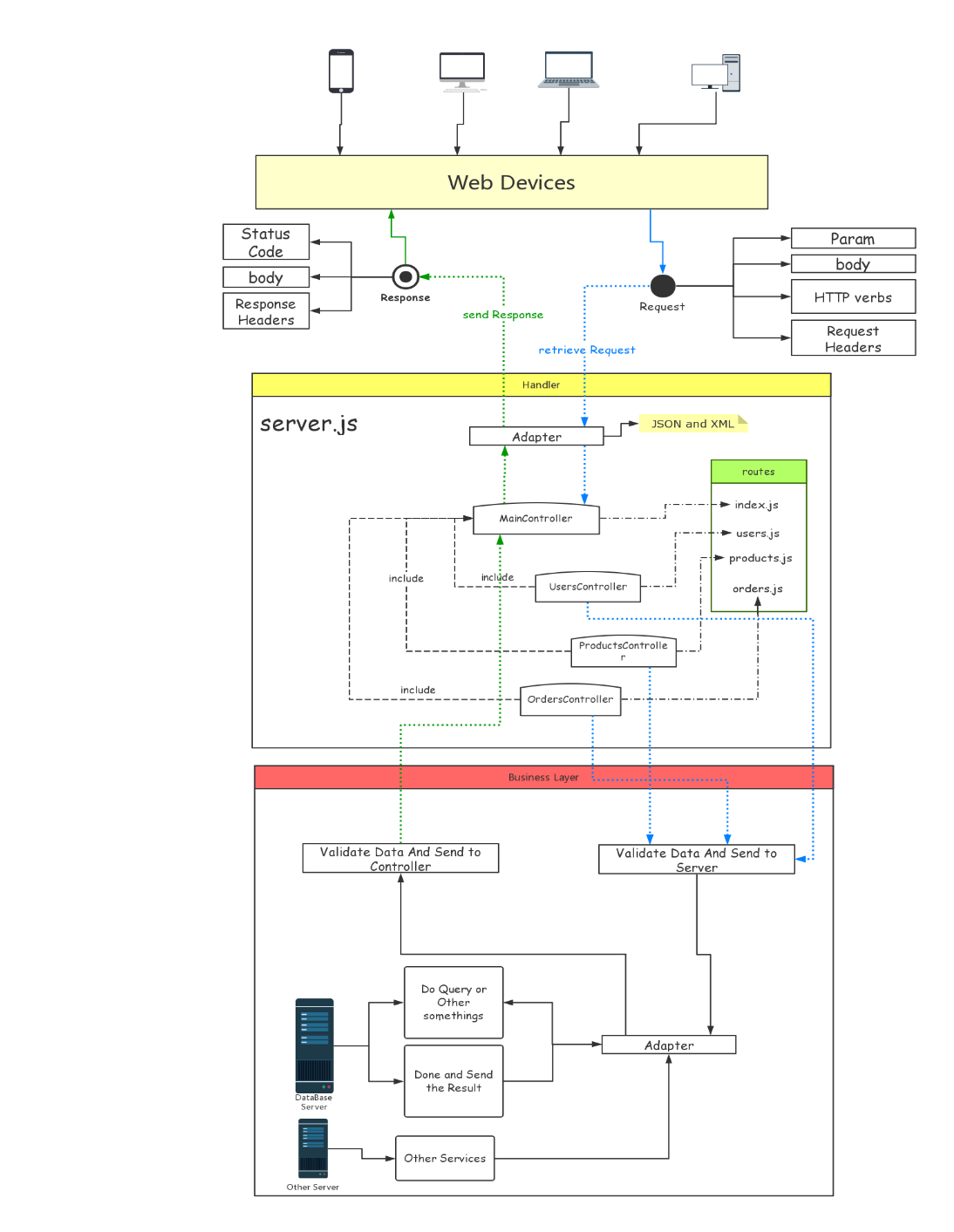


图 ‑1架构设计分层示意图

表现层：用于展示系统的业务信息以及接受用户输入信息。

控制层：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，同时接收业务逻辑数据访问层处理结果，指定相应的表现层 jsp 页面展现需求数据，实现页面跳转与信息现实。

业务逻辑数据访问层：业务逻辑数据访问层为系统的核心层，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。



## 模块简介

BrandiTravel主要包括的模块为两部分：前台展示系统模块、BrandiTravel后台管理模块，如图 2‑2所示。

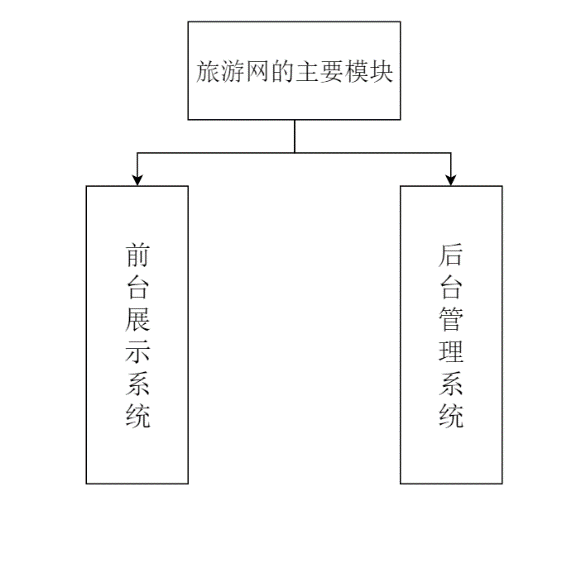


图 ‑2 BrandiTravel主要模块图

前台展示业务主要是为了给用户提供旅游路线展示，旅游路线选择，个人信息填写，进行以个人信息为主的商品购买流程。其中详细的功能包括了用户个人的注册和登录。登录后，用户可以进行路线页面浏览访问，并且可以针对自己喜欢中意的旅游路线进行添加购物车功能。在购物车界面或者在路线详情界面直接可以对路线进行支付信息。其中包括路线支付和出游人员信息填写。整个前台展示系统以围绕用户展开，由商品路线为承接，可以让用户进行下一步操作。

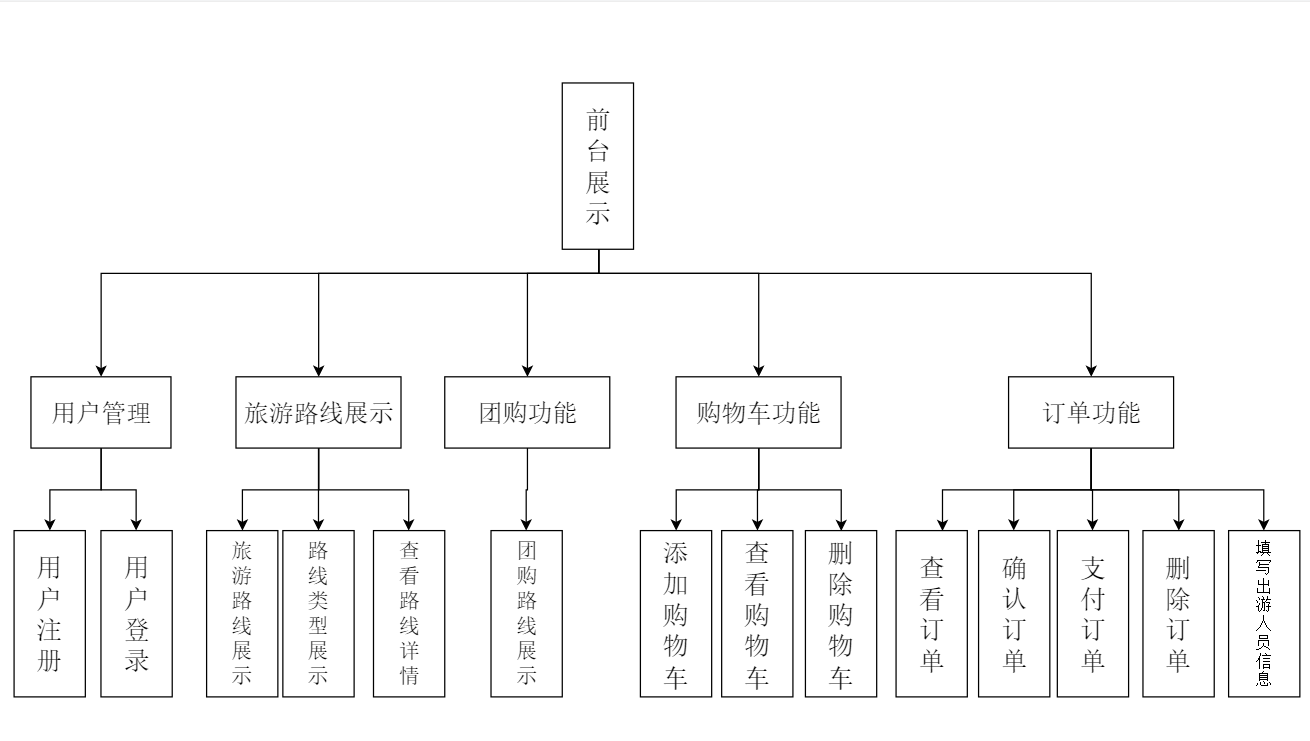


图 ‑3前台展示业务系统功能模块图

1. 用户进入系统首页后，系统即显示登录界面，用户登录后系统可判断用户类型进而提供不同的服务操作。如果用户没有用户名及密码，可以直接进行系统注册。
2. 用户在登录成功后，直接进入主页面的路线浏览界面，其中包括了路线的类型预览，路线人气排行预览等相关页面。
3. 在用户选择好相应路线类型后，就可以直接进入该类型了解更多的不同路线。用户可以直接看见该路线的简介，然后可以直接通过对商品路线的图片和简介了解该路线，用户也可以点击该路线，进入路线详情中了解更多，包括了出行方式，出行时间，推荐出行人数等等，或是即将开放的评论区。用户可以在此界面进行当机立断的结账功能，或是再三考虑添加入购物车的功能。
4. 用户在主页面下还可以浏览到团购路线信息。该信息的路线不包括所有路线，只是仅仅对一些可以进行折扣路线优惠的团购。当用户对其余路线有些许犹豫但是团购路线当中存在中意路线，用户可以进行团购路线选择。
5. 用户在路线展示界面偶读浏览大致后，进入购物车界界面再次对即将确定的路线进行确认，其中包括了对购物车内的一个商品路线的删除或结账操作，当用户移除了一个商品可以重新回到那个商品路线详情里，再次将其添加回购物车。
6. 当用户们已经确定了一条路线并提交支付了，就进入填写订单界面进行订单的填写。包括了对出行人员的信息填写。提交了相应填写内容后进入订单内容确认，保证了用户可能会出现的填写错误。当所有确认订单工作完成后，跳转到支付页面进行订单支付。支付订单后，系统会将用户的填写信息与扣款信息上传到服务器，服务器处理后递交成功请求，用户就可以在订单列表等待相应的出行时间等待美好的旅途。

后台管理模块如图 2‑4所示，主要为系统管理人员提供服务，其功能包括了对路线类型的管理，路线信息的管理，团购路线信息的管理。。

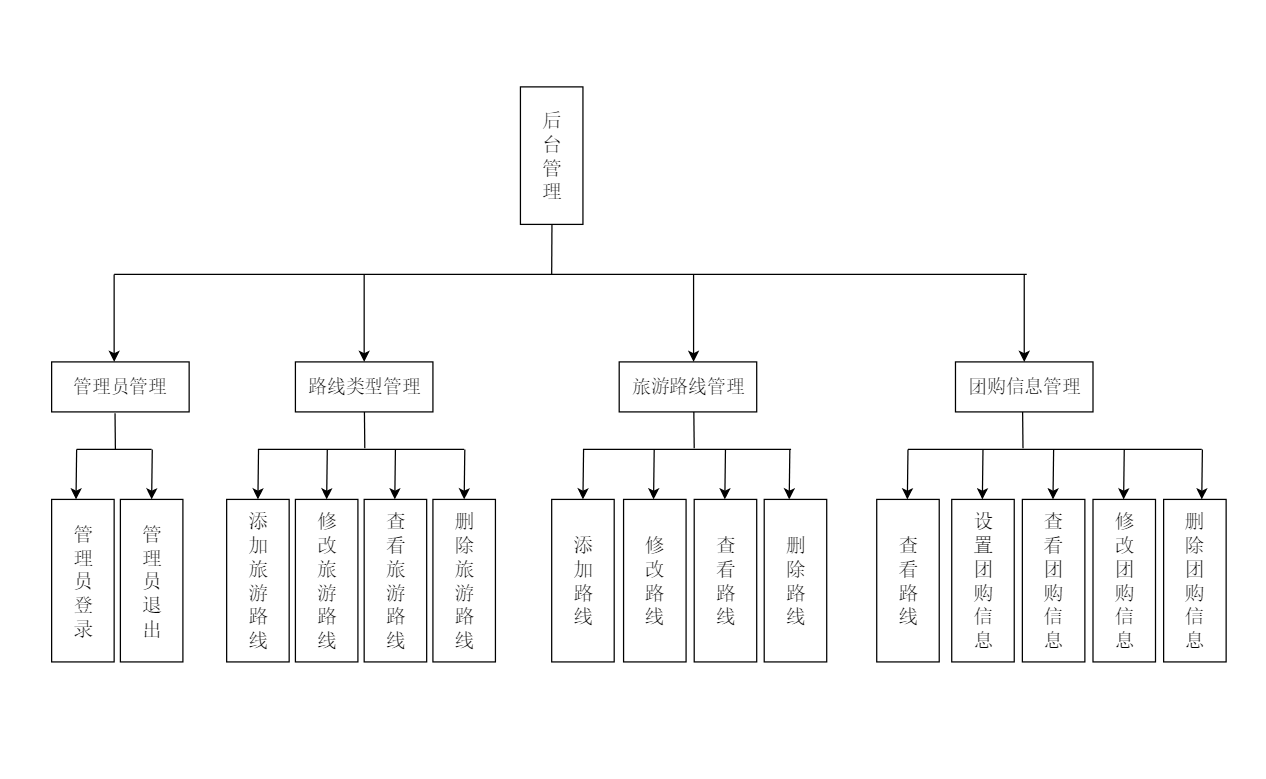


图 ‑4后台管理系统功能模块图

1. 管理员通过账户密码的方式登录，提交了登录信息后，系统授权登录，管理员进入了后台管理系统界面，然后直接进行相应信息的增加，更改，删除操作。
2. 管理登录成功后，可以进入管理路线信息界面，在该界面中，管理员可以直接对路线信息进行更改，包括其出游时间，出游方式，价格等可以显示在用户界面的路线详情内容。
3. 管理员登录成功后，可以进入管理路线类型界面，对路线的类型进行修改，包括了对境内游，境外游等多种类型的路线类型修改，也可以在相关路线类型上操作，将部分路线从该类型中移除。

# 前台展示业务模块详细设计

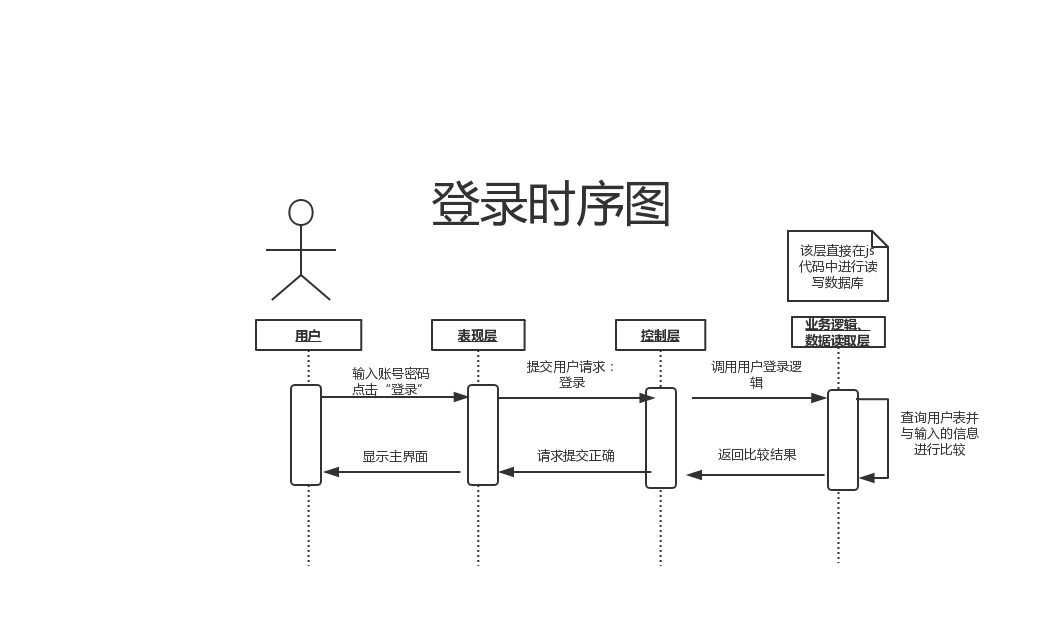
前台基本业务模块主要包括用户登录模块、用户注册模块、首页旅游路线展示，旅游路线类型展示，旅游路线详情展示，团购路线信息展示，购物车功能，订单功能。



## 用户登录模块

用户登录模块系统内部的时序示意图如图 3‑1所示。

图 ‑1用户登录时序示意图

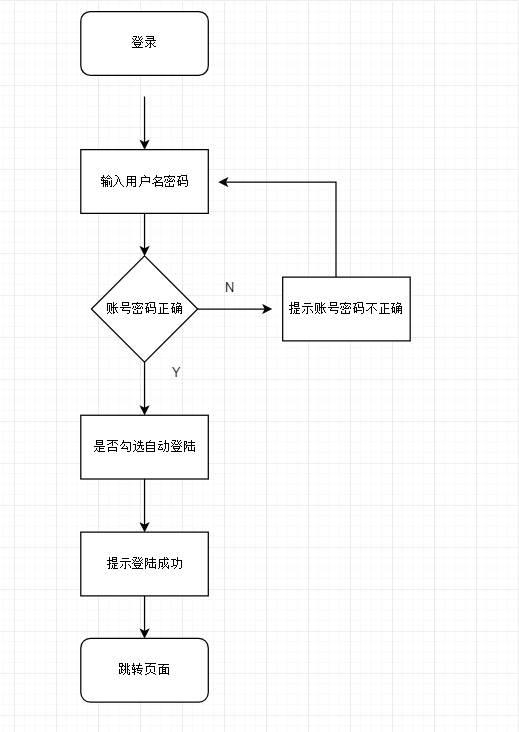




### 表现层

登录模块的表现层主要完成不同用户的登录功能，其中在登录页面要求用户输入账号密码的基本信息，确认后页面给出响应消息，登录成功进入网站首页，登录失败提示账户名或者密码错误。表现层的流程图如图 3‑2所示。

图 ‑2用户登录行为流程图



### 控制层

登录模块的控制层负责接受来自表现层的用户输入，同时调用登入模块的业务逻辑接口，将用户名与密码等用户关键信息传递到业务逻辑数据访问层进行判定。等到业务逻辑处理完成之后，将来自业务逻辑数据访问层的相应信息传到表现层，并决定显示页面。

表 3‑1用户登录展示控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 主控制组件 | 子控件单元 | 说明 |
| 用户登录 | UserController | UserLogin.js | 登录页面页面显示 |

### 业务逻辑数据访问层

登录模块的业务逻辑数据访问层主要完成对用户登录逻辑的判定，同时调用登录模块的业务逻辑接口。比如：用户到登录是输入的用户名是否存在、密码是否正确。若存在错误则返回填写界面重新填写，若正确则由数据访问储存到数据库当中。

表 3‑2业务逻辑数据访问层

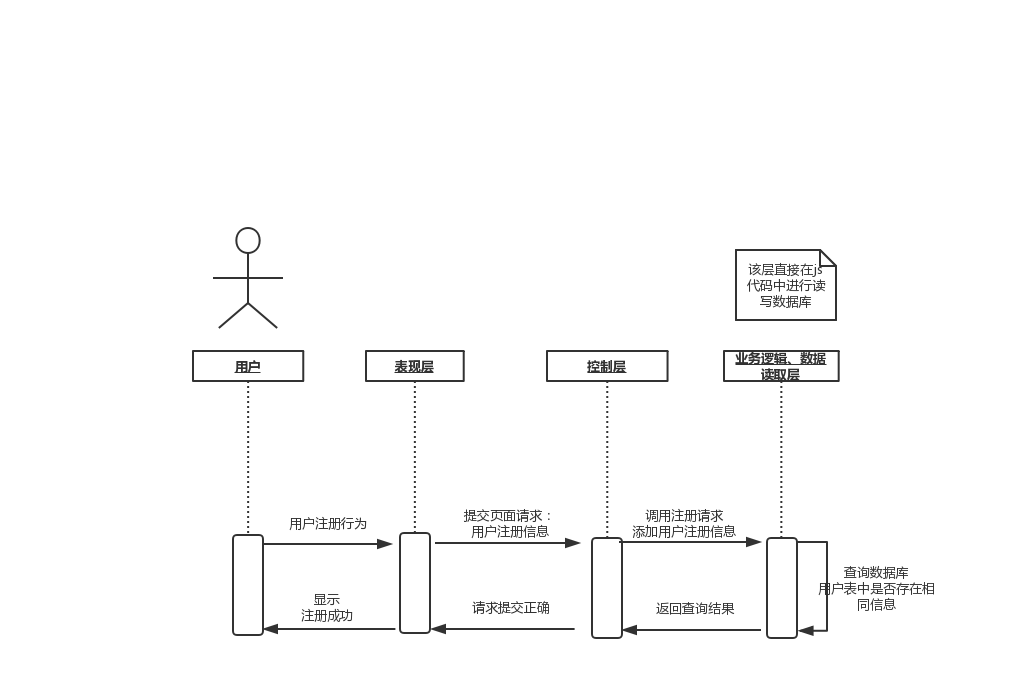
|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 业务逻辑处理单元 | 调用说明 | 说明 |
| 用户登录 | People.js | 调用数据库 | 返回给 UserController |



## 用户注册模块

在注册模块时，系统内部的时序示意图如图 3‑3 所示。

图 ‑3用户注册流程示意图







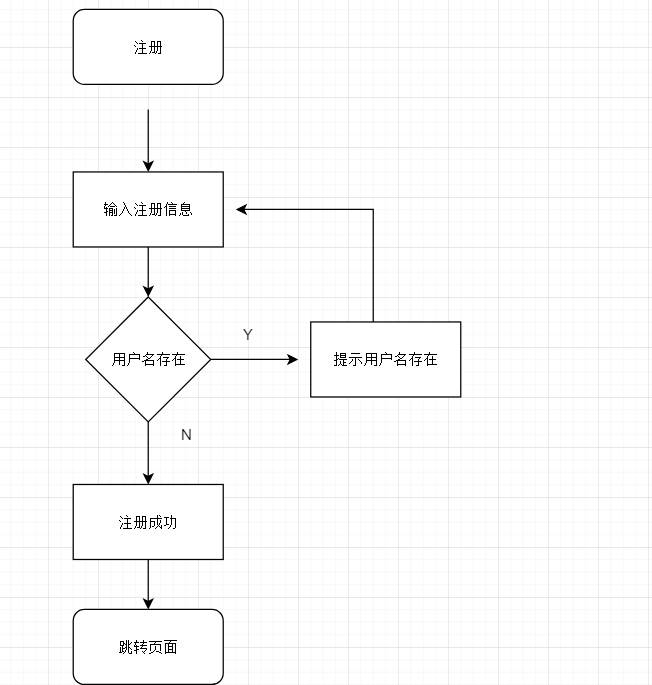




### 表现层

注册模块的表现层主要完成不同用户的注册功能，其中在注册页面要求用户输入要注册的账号和密码，点击确认注册后自动登录跳转到主页，注册失败提示注册信息有误。表现层的流程图如用户注册流程图如图 3‑4所示。

用户注册流程图如图 3‑4所示。





### 控制层

注册模块通过UserRegiseter.js控制组件输入，同时调用注册模块的业务逻辑接口，将用户名与密码等用户关键信息传递到业务逻辑数据访问层进行判定。等到业务逻辑处理完成之后，将来自业务逻辑数据访问层的相应信息传到表现层，并决定显示页面。

表 3‑3旅游路线详情展示控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 主控制组件 | 子控件单元 | 说明 |
| 用户注册 | UserController | UserRegister.js | 注册页面页面显示 |

### 业务逻辑数据访问层

注册模块的业务逻辑数据访问层主要完成对用户注册逻辑的判定，同时调用注册模块的业务逻辑接口。比如：用户注册模块是对输入的用户名是否合法、密码是否合法以及确认密码与密码是否相同的判定。若存在相同提交信息则返回已注册信息， 若没有则上传数据到数据库进行保存。

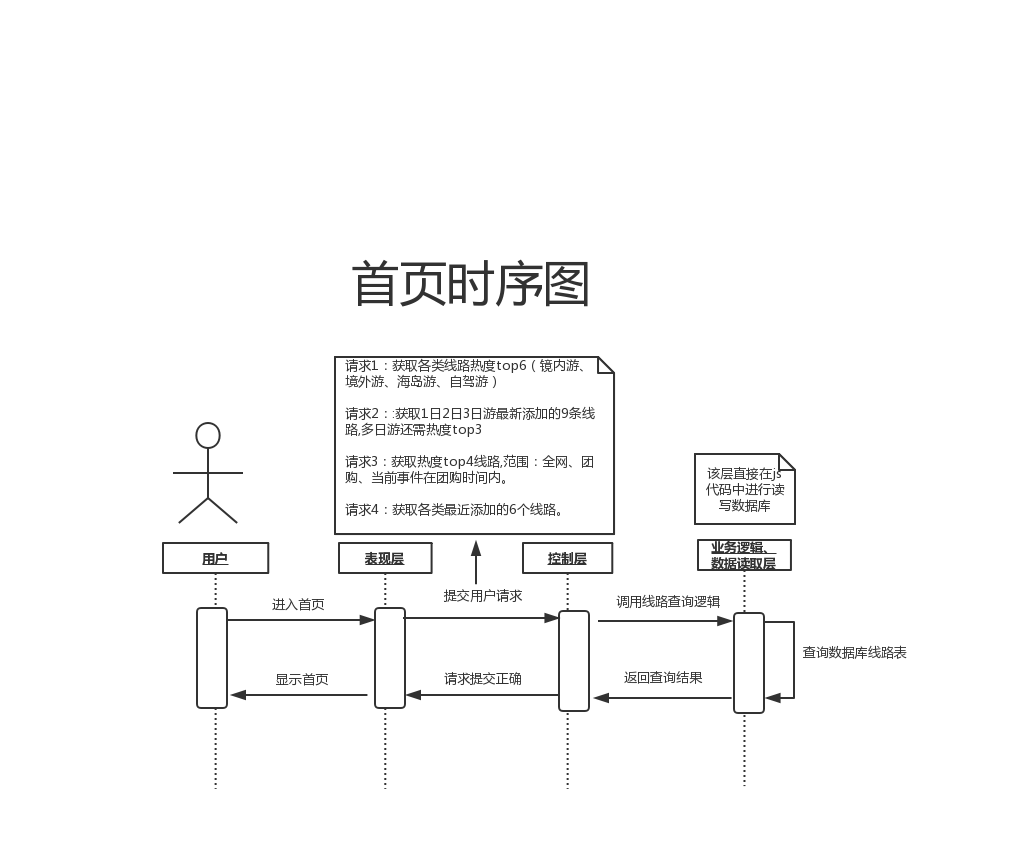
表 3‑4业务逻辑数据访问层

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 业务逻辑处理单元 | 调用说明 | 说明 |
| 用户注册 | People.js | 调用数据库 | 返回给 UserController |

## 首页旅游路线展示模块

首页旅游路线展示示意图如图 3‑5所示。

图 ‑5 时序示意图

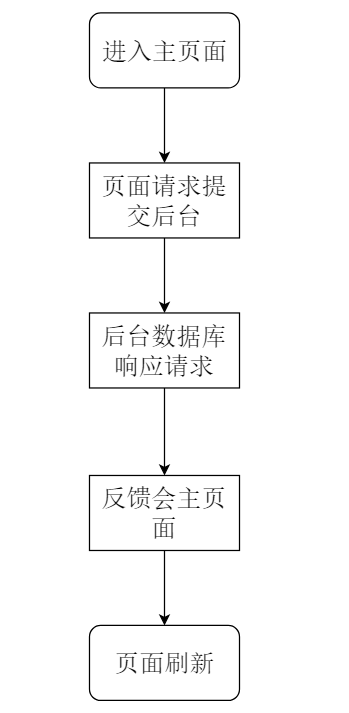




### 表现层

首页旅游路线浏览模块的表现层主要完成用户与后台数据访问功能。实现了将后台图片与数据通过用户对网页的访问请求的反馈，让用户直截了当地可以浏览大致的旅游路线，其中表现层的接口用了REST API的复合接口，从其中连接Request，有效地将信息返回刷新到主页面上。表现层的流程图如图 3‑6所示。

图 ‑6浏览主页流程示意图



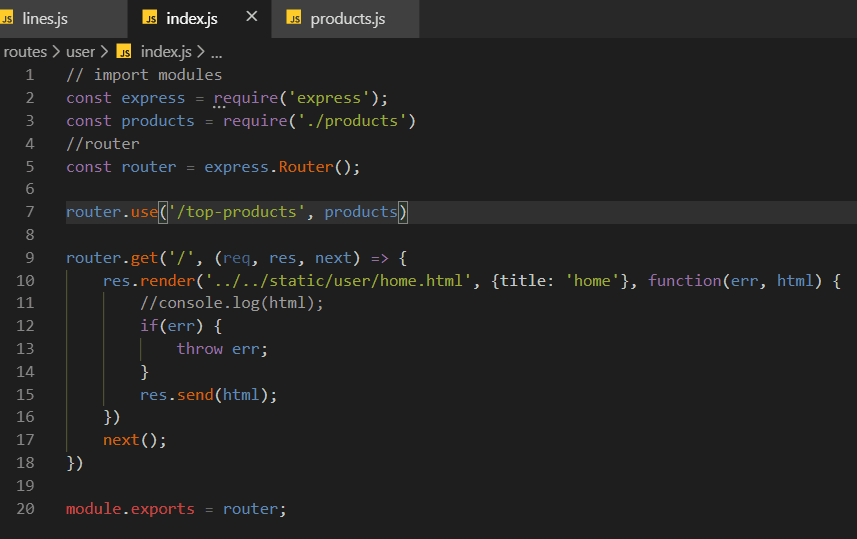
### 控制层

首页旅游路线展示模块的控制层负责接受来自 index.js 的用户输入，同时首页旅游路线模块的业务逻辑接口，查询用户列表并返回带有用户名称的写信页面。用户访问改主页面并提交访问请求，有业务逻辑数据访问层响应并返回请求，显示页面。首页旅游路线展示模块控制层列表见表 3‑5所示。

表 3‑5首页旅游路线展示模块控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 主控制组件 | 子控件单元 | 说明 |
| 首页旅游  路线展示 | MainController | Index.js | 主页面显示 |

在控制层中 index.js 的主要属性与方法如下所示：



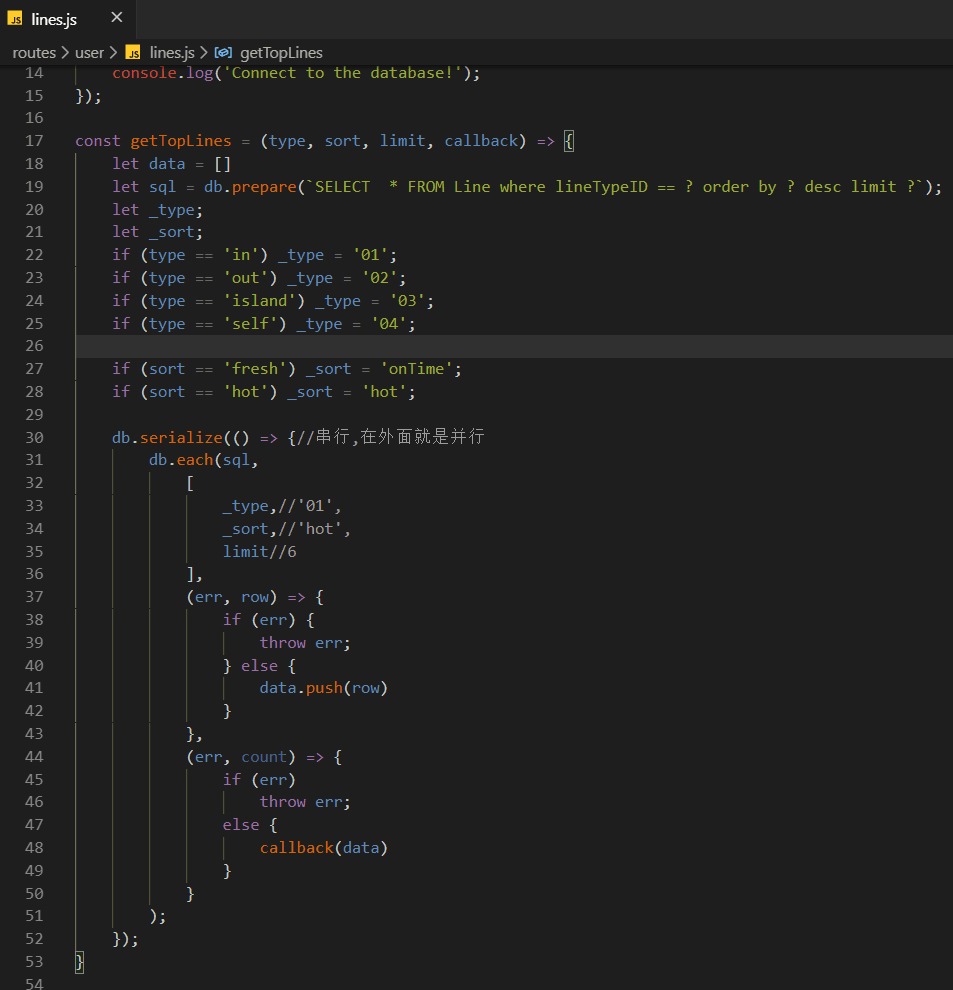
### 业务逻辑数据访问层

首页旅游路下你展示模块的业务逻辑数据访问层主要完成对路线展示逻辑的判定，同时将index.js子控制单元的相应参数值传给Line.js业务逻辑处理单元进行函数调用，进行数据库查询，再由结果回调，返回查询结构

表 3‑6业务逻辑数据访问层

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 业务逻辑处理单元 | 调用说明 | 说明 |
| 首页路线展示 | Line.js | 调用数据库 | 返回给 MainController |

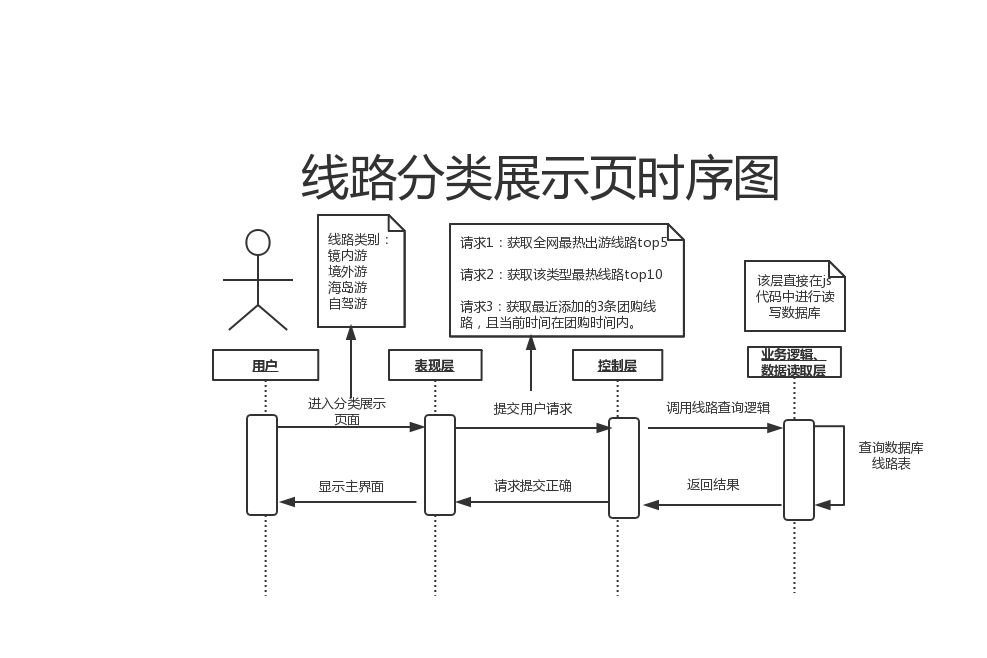
图 ‑7 业务逻辑数据访问层Line.js



## 旅游路线类型展示模块

选课模块系统内部的相应操作示意图如图 3‑8所示。

图 ‑8时序示意图

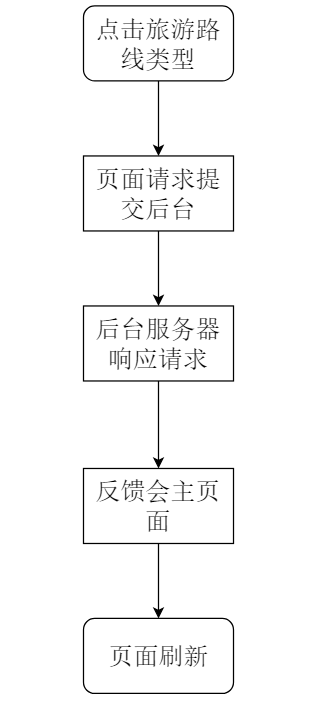




### 表现层

旅游路线类型展示模块是针对用户对不同类型的旅游路线选择的界面，用户可以对自己向往中意的类型进行不同路线选择。用户进入到主页面选择好的相应类型的模块，整个模块功能是将不同类型的旅游路线通过由后台数据库的分类整合，再根据用户对所需要的的旅游类型网页访问，回调不同类型的Request接口，在响应用户不同界面的旅游路线类型。表现层的流程图如图 3‑9所示。

图 ‑9旅游路线类型示意图



### 控制层

旅游路线类型展示页面通过product.js子控制单元进行页面跳转。等到相应参数值传递给业务逻辑处理完成之后，将来自业务逻辑数据访问层的相应信息传到表现层，并决定显示页面。

表 3‑7旅游路线类型展示控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 主控制组件 | 子控件单元 | 说明 |
| 旅游路线类型  路线展示 | MainController | LineType.js | 旅游路线类型页面显示 |

### 业务逻辑数据访问层

选课模块的业务逻辑数据访问层主要完成对用户条件筛选的逻辑判定，同时调用用户搜索的业务逻辑连接。比如：搜索用户名成功刷新用户界面,条件筛选刷新用户界面等。选课模块业务逻辑数据访问层列表如图4-18所示。

表 4-18 旅游路线类型展示模块业务逻辑数据访问层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 业务逻辑处理单元 | 调用说明 | 说明 |
| 旅游路线类型展示 | Line.js | 调用数据库 | 返回给 LineController |

在本模块的业务逻辑数据访问层调用了randUsrs.java 接口，同时通过 randUsrsImpl.java 实现该接口。

randUsrsImpl.java实现了 randUsrs.java 接口， 同时需要调用数据访问层的UsrsMapper.java和AddressMapper.java，来实现对数据访问层接口的调用。

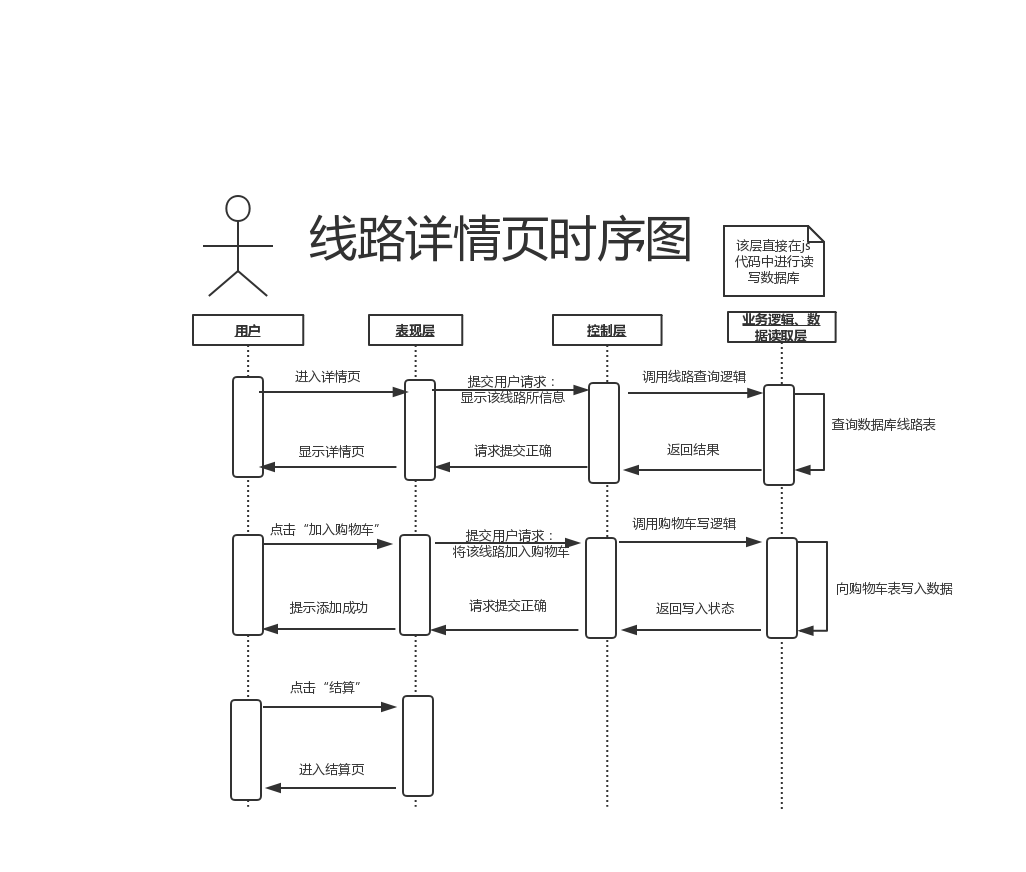
randUsrsImpl.java 主要实现属性与方法:

表 3‑8用户搜索模块数据访问层列表

## 旅游路线详情展示模块

旅游路线详情的时序示意图如图 3‑10所示。

图 ‑10时序示意图

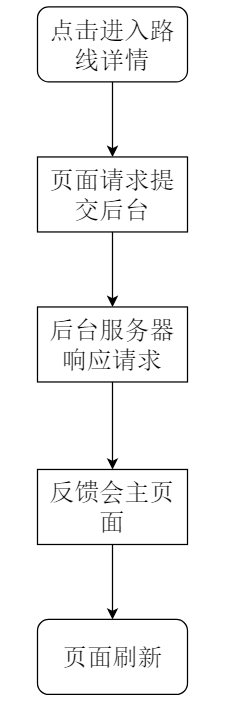




### 表现层

旅游路线详情展示模块主要为用户们对某一路线详情概况的描述功能，包括了对一条路线的出行价格，出行天数，出行方式，出行推荐人数等等功能性展示。该模块也是由用户对路线的简介图片或文字点击，通过文字图片的链接跳转到改路线详情界面，网页请求通过表现层Request接口传到后台服务器，服务器再将请求返回，在页面刷新后将该路线的详细情况展示出来，供用户浏览并做选择。

图 ‑11旅游路线详情流程示意图



### 控制层

旅游路线详情展示是等待用户从表现层通过文字或图片的点击，然后跳转至该路线详情界面。就是由点击事件的请求发送到LineController主控制组件当中，传值到子控制组件LineDetails。Js中，再等到业务逻辑处理完成之后，将来自业务逻辑数据访问层的相应信息传到表现层，并决定显示页面。登录模块控制层列表见表 3‑9所示。

表 3‑9旅游路线详情展示控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 主控制组件 | 子控件单元 | 说明 |
| 旅游路线详情  路线展示 | LineController | LineDetails.js | 旅游路线详情页面显示 |

### 业务逻辑数据访问层

旅游路线详情模块的业务逻辑数据访问层主要是接受来自LineDetails.js的参数值。同个单个的参数值，再调函数进行用相关路线的数据库查询，包括了从表到资源定位符，再将结果返回到控制组件并返回到表现层。旅游路线详情模块业务逻辑数据访问层列表如表 3‑10所示。

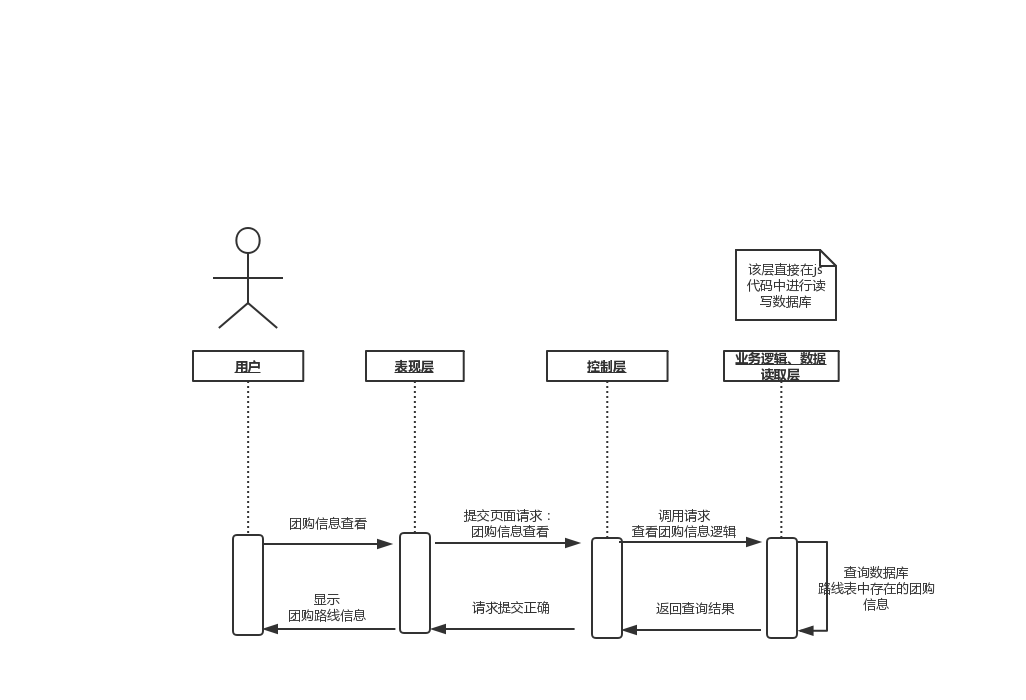
表 3‑10旅游路线详情展示模块业务逻辑数据访问层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 业务逻辑处理单元 | 调用说明 | 说明 |
| 旅游路线详情展示 | Line.js | 调用数据库 | 返回给 LineController |

## 团购路线展示模块

团购路线展示的时序示意图如图 3‑12时序示意图 所示。

图 ‑12时序示意图

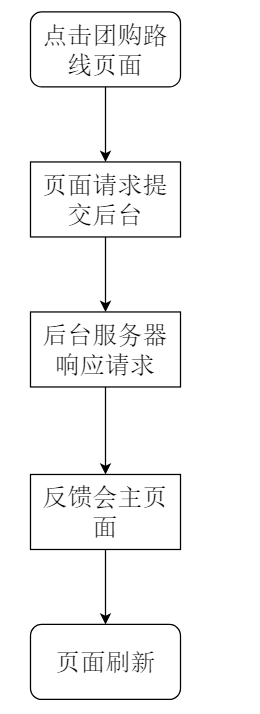




### 表现层

团购信息展示模块主要是用户可以通过查看团购路线情况，并根据自己的需求意愿进行选择，但是用户只能选择团购路线中已存在的路线，并不是每一条路线都有团购优惠。用户查看团购路线时，是将其请求数据发送到后台服务器，然后由服务器根据数据库中路线表当中查询存在的团购路线，再返回查询结果，按照网页排版进行用户页面展示。表现层的流程图如图 3‑13团购旅游路线流程示意图所示。

图 ‑13团购旅游路线流程示意图



### 控制层

团购路线信息展示模块通过用户在来自表现层的点击事件对ProductController主控制组件的输入，并由其传值给该主控制组件的子控制组件Group.js，将展示部分的参数值传递到业务逻辑数据访问层进行判定。等到业务逻辑处理完成之后，将来自业务逻辑数据访问层的相应信息传到表现层，并决定显示页面。团购路线信息展示模块控制层列表见表 3‑11旅游路线详情展示控制层列表所示。

表 3‑11旅游路线详情展示控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 主控制组件 | 子控件单元 | 说明 |
| 团购路线信息  展示 | Productontroller | Group.js | 团购路线信息页面显示 |

### 业务逻辑数据访问层

团购路线信息展示模块的业务逻辑数据访问则是通过对业务逻辑处理单元的调用实现从后台至。团购信息模块业务逻辑数据访问层列表如表 3‑12旅游路线详情展示模块业务逻辑数据访问层列表所示。

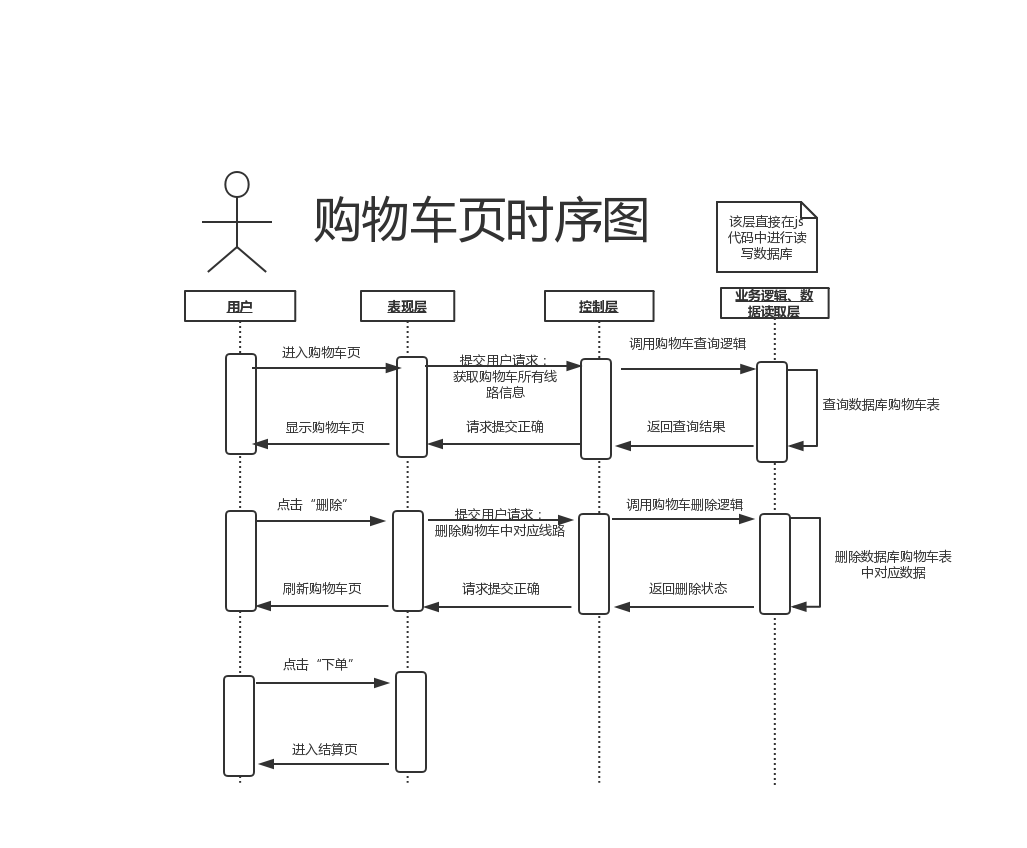
表 3‑12旅游路线详情展示模块业务逻辑数据访问层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 业务逻辑处理单元 | 调用说明 | 说明 |
| 团购路线信息展示 | Product.js | 调用数据库 | 返回给 LineController |

## 购物车功能模块

在发布活动模块时，系统内部的时序示意图如图 3‑14时序示意图所示。

图 ‑14时序示意图

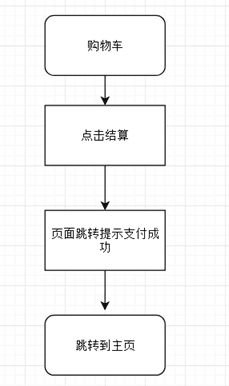




### 表现层

购物车功能模块主要为用户服务购物商品清单，用户可以将有意愿的中意的商品路线用过购物车进行存储，存储模式通过堆栈方式存入用户信息后台数据当中，当用户对购物车进行移除操作时，该请求也会传递服务器并依次做出出栈操作移除掉用户想要删除的商品路线。购物车功能方便用户在取舍不同路线时，有一个缓冲。当用户决定好一个路线或多个路线时，可以通过购物车功能中提供的结账功能，通过点击该按钮可以直接跳转至订单界面。表现层对应的流程图。

图 ‑15购物车流程示意图



### 控制层

购物车功能模块是由来自用户从表现层对购物车按钮店家并跳转至下一界面的鼠标点击时间相应，再从其ProductController主控制组件当中的Group.js子控制组件进行传值输入到业务逻辑数据访问层当中，并由业务逻辑数据访问层中的Product.js业务逻辑处理单元进行函数调用，等到业务逻辑处理完成之后，将来自业务逻辑数据访问层的相应信息传到表现层，并决定显示页面。

购物车添加则是通过来自网页上，对购物车添加功能的点击事件，并将其点击的路线编号通过ShoppingCarAdd.js子控制组件传值到Product.js处理单元当中进行业务逻辑排序，依次入栈的方式进入购物车当中。由业务逻辑数据访问层进行数据响应并返回。

购物车删除由ShoppingCarDelete.js进行数据传值，再通过Product.js业务逻辑处理单元对数据库库的访问并执行操作，将关联用户信息的数据库购物车表进行对应商品的删除，最后返回结果到主控制组件Productontroller，再传递回表现层用户可以显示界面。

购物车支付则是由ShoppingCarPay.js直接进入到页面跳转单元，进入页面支付后用户交互的数据传给OrderFill.js当中，进行订单操作然后按照订单流程进行。

购物车模块控制层列表见表 3‑13 所示。

表 3‑13购物车功能控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 主控制组件 | 子控件单元 | 说明 |
| 购物车添加功能 | Productontroller | ShoppingCarAdd.js | 购物车添加 |
| 购物车删除功能 | Productontroller | ShoppingCarDelete.js | 购物车删除 |
| 购物车结账功能 | Productontroller | ShoppingCarPay.js | 购物车结账 |

### 业务逻辑数据访问层

购物车功能模块的业务逻辑数据访问层主要完成用户对购物车当中的操作进行逻辑的判定，包括对购物车的添加，移除，结账或商品详情页面跳转操作，这些函数都来自Product.js业务逻辑处理单元的调用，同时调用对应的数据库，并查询存在结果和返回。购物车功能模块业务逻辑数据访问层列表如表 3‑14 所示。

表 3‑14购物车功能模块业务逻辑数据访问层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 业务逻辑处理单元 | 调用说明 | 说明 |
| 购物车添加功能 | Product.js | 调用数据库 | 返回给 ProductController |
| 购物车删除功能 | Product.js | 调用数据库 | 返回给 ProductController |
| 购物车结账功能 | Product.js | 调用数据库 | 返回给 ProductController |

## 订单功能模块

订单功能的时序示意图如图 3‑16历史订单时序示意图，图 3‑17订单确认结算时序示意图所示。

图 ‑16历史订单时序示意图

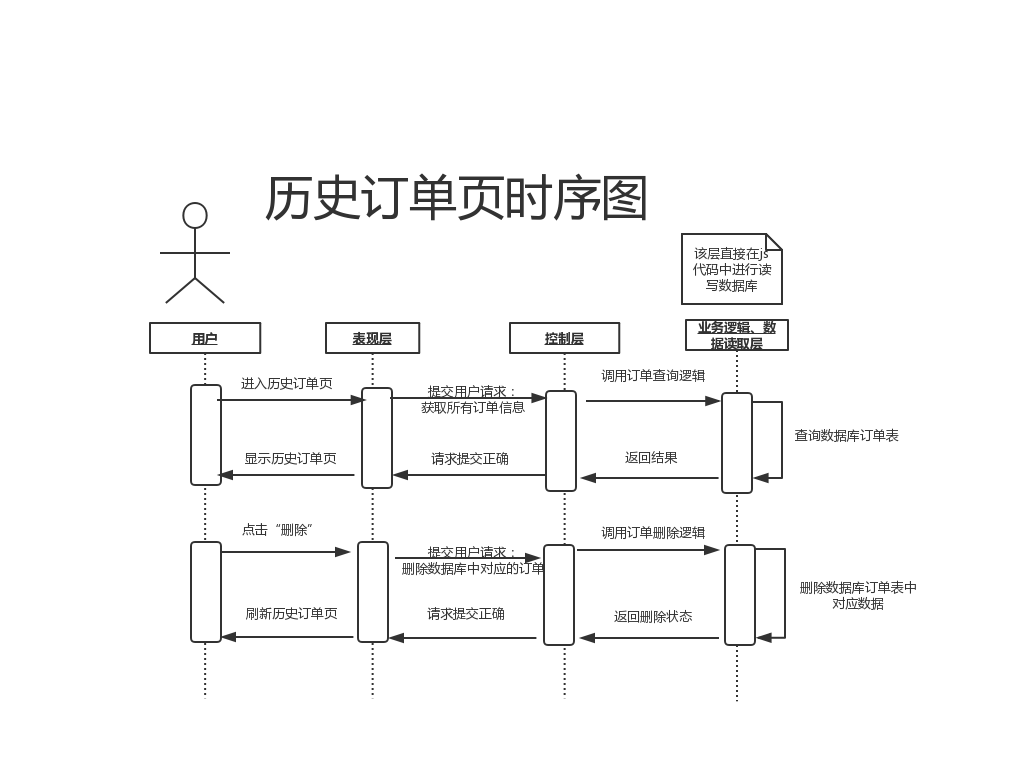
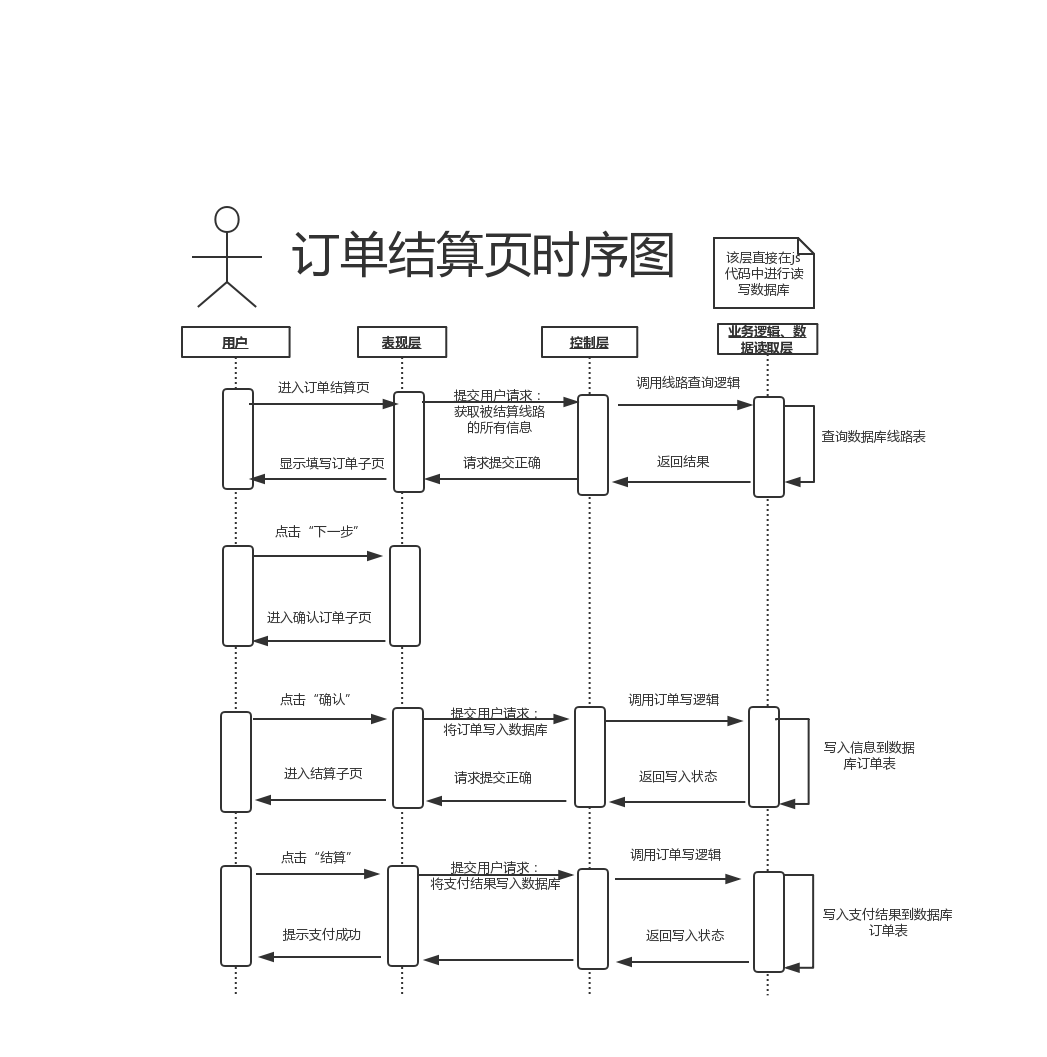


图 ‑17订单确认结算时序示意图







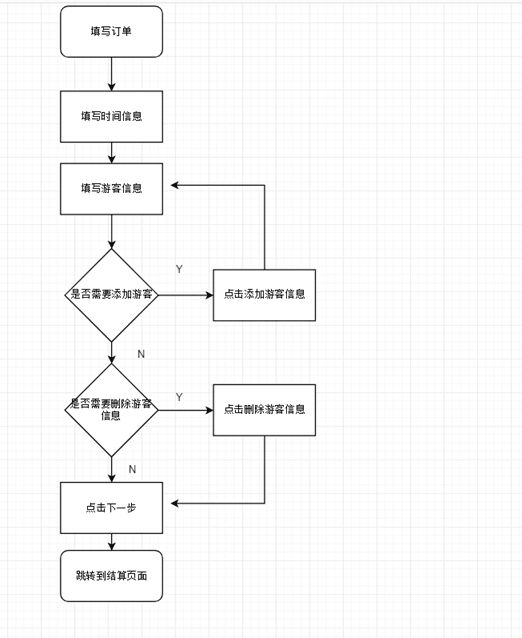




### 表现层

订单功能模块主要是为了使用户可以在确定好路线并且对该路线进行相应信息填写和支付操作，在已出游的订单列表中存在可以进行一些订单的删除操作。订单填写主要是由用户在订单页面填写文本信息并上传到后台服务器，由服务器保存相应数据并把跳转页面的请求回传到用户界面。表现层对应的流程图。

图 ‑18订单功能流程图



### 控制层

订单功能模块的控制层来自表现层中用户对订单界面的网页跳转，进入该界面后，用户进入订单填写界面，通过对填写的文本输入，将文本输入的所有内容由主控制组件Productontroller指向输入，并由子控制组件OrderFill.js传值到业务逻辑数据访问层，存入到数据库里。

订单支付则由主控制组件Productontroller中的子控制组件OrderPay.js针对用户跳转的支付环境进行确认，并其支付信息确认传值到业务逻辑数据访问层当中。

订单查看则是在通过对订单列表的鼠标点击事件，为显示已事件排序的订单列表，子控制组件OrderCheck.js对以往订单进行时间传值，由业务逻辑数据访问层进行函数调用排序，再返回相应结果并按照表现层排版。

订单删除则是直接由OrderDelete.js对用户点击的相应订单进行直接到业务逻辑数据访问层中对排序数组的删除操作，删除历史订单列表并相关联到用户信息，实现在用户界面可以看见的删除操作。

订单功能模块控制层列表见表 3‑15订单功能展示控制层列表 所示。

表 3‑15订单功能展示控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 主控制组件 | 子控件单元 | 说明 |
| 订单填写 | Productontroller | OrderFill.js | 订单填写页面显示 |
| 订单支付 | Productontroller | OrderPay.js | 订单支付页面显示 |
| 订单查看 | Productontroller | OrderCheck.js | 订单列表页面显示 |
| 订单删除 | Productontroller | OrderDelete.js | 订单删除功能 |

### 业务逻辑数据访问层

订单功能的所有实现函数均在业务逻辑数据访问层中的业务逻辑处理单元进行相关函数调用。包括了简单的增删改查功能，和直接对订单出行人员的信息文本存档至数据库。所有的操作完成后，等待从数据访问的数据库响应，并返回响应的结果，传递到主控制单元当中，再显示回表现层。订单功能模块业务逻辑数据访问层列表如表 3‑16 所示。

表 3‑16订单功能展示模块业务逻辑数据访问层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 业务逻辑处理单元 | 调用说明 | 说明 |
| 订单填写 | Product.js | 调用数据库 | 返回给 ProductController |
| 订单支付 | Product.js | 调用数据库 | 返回给 ProductController |
| 订单查看 | Product.js | 调用数据库 | 返回给 ProductController |
| 订单删除 | Product.js | 调用数据库 | 返回给 ProductController |

# 后台管理系统的详细设计

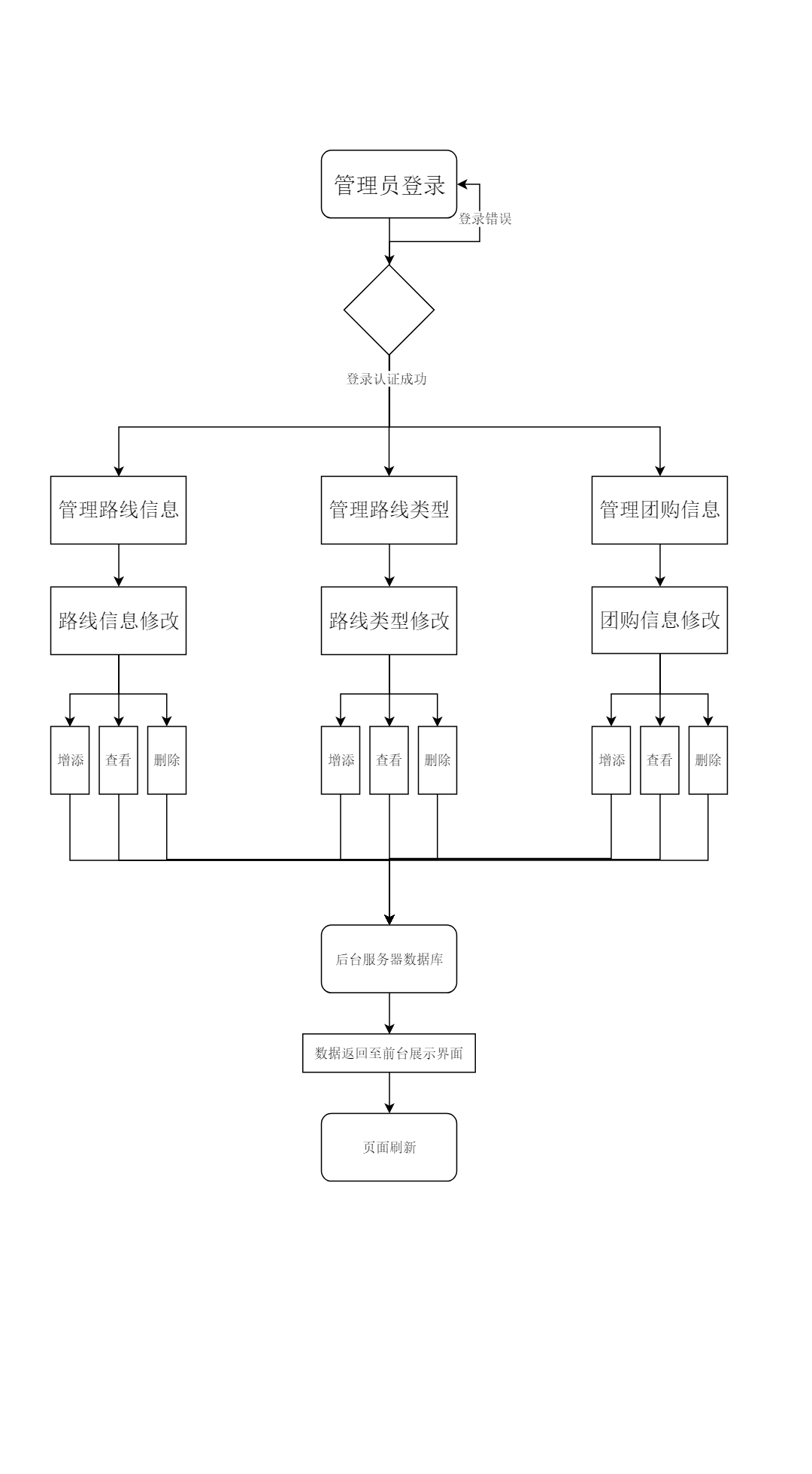
主要是对系统信息的管理，包括解禁，封禁用户，以及活动和交友论坛模块文章和评论的删除与恢复。



## 后台管理层模块表现层

后台管理模块的表现层主要由管理员对于网站的后台的路线信息管理，路线类型管理，团购路线信息管理。管理的内容包括了管理员对于路线信息，路线类型，团购信息的增删改查。管理员的每次操作都会在其登出时候进行后台保存，并且在往后推出固定维护时间进行数据维护，将数据库的修改内容通过控制接口返回到用户页面，保证用户可以看见管理员在修改后的信息。表现层对应的流程图。

图 ‑1管理员后台管理模块流程图



## 后台管理模块控制层

后台管理模块的控制层主要是控制页面的跳转与将从业务逻辑数据访问层获取到的数据传送到表现层。所有后台的控制模块的主控制组件都为ManageController，然后针对每一个不同功能的函数使其作为子控制组件。

表 4‑1旅游路线详情展示控制层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 主控制组件 | 子控件单元 | 说明 |
| 路线信息添加 | ManageController | LineMessgeAdd.js | 实现路线信息添加 |
| 路线信息修改 | ManageController | LineMessgeChange.js | 实现路线信息修改 |
| 路线信息查看 | ManageController | LineMessgeCheck.js | 实现路线信息查看 |
| 路线信息删除 | ManageController | LineMessgeDelete.js | 实现路线信息删除 |
| 路线类型添加 | ManageController | LineTypeAdd.js | 实现路线类型添加 |
| 路线类型修改 | ManageController | LineTypeChange.js | 实现路线类型修改 |
| 路线类型查看 | ManageController | LineTypeCheck.js | 实现路线类型查看 |
| 路线类型删除 | ManageController | LineTypeDelete.js | 实现路线类型删除 |
| 团购信息添加 | ManageController | GroupMessageAdd.js | 实现团购信息添加 |
| 团购信息修改 | ManageController | GroupMessageChange.js | 实现团购信息修改 |
| 团购信息查看 | ManageController | GroupMessageCheck.js | 实现团购信息查看 |
| 团购信息删除 | ManageController | GroupMessagDelete.js | 实现团购信息删除 |

## 后台管理模块业务逻辑数据访问层

后台管理模块的业务逻辑数据访问层将调用底层的数据访问层将相应的信息获取，并传递给控制层。由Manage.js作为业务逻辑处理单元，并且每一个子控制组件都是传值进入该业务逻辑处理单元进行最基本的增，删，改，查函数操作。具体见表 4‑2。

表 4‑2旅游路线详情展示模块业务逻辑数据访问层列表

|  |  |  |  |
| --- | --- | --- | --- |
| 事件 | 业务逻辑处理单元 | 调用说明 | 说明 |
| 路线信息添加 | Manage.js | 调用数据库 | 返回给 ManageController |
| 路线信息修改 | Manage.js | 调用数据库 | 返回给 ManageController |
| 路线信息查看 | Manage.js | 调用数据库 | 返回给 ManageController |
| 路线信息删除 | Manage.js | 调用数据库 | 返回给 ManageController |
| 路线类型添加 | Manage.js | 调用数据库 | 返回给 ManageController |
| 路线类型修改 | Manage.js | 调用数据库 | 返回给 ManageController |
| 路线类型查看 | Manage.js | 调用数据库 | 返回给 ManageController |
| 路线类型删除 | Manage.js | 调用数据库 | 返回给 ManageController |
| 团购信息添加 | Manage.js | 调用数据库 | 返回给 ManageController |
| 团购信息修改 | Manage.js | 调用数据库 | 返回给 ManageController |
| 团购信息查看 | Manage.js | 调用数据库 | 返回给 ManageController |
| 团购信息删除 | Manage.js | 调用数据库 | 返回给 ManageController |

# 公共部分模块详细设计



## 公共页面

公共页面的基本界面信息即相应的功能描述如表 5‑1 所示。

表 5‑1页面表现层部分设计

|  |  |  |
| --- | --- | --- |
| 文件名称 | Html | 说明 |
| 用户登录 | test.html | 显示**用户登录**界面 |
| 主页显示 | home.html | 显示**主页显示**界面 |
| 路线类型 | Out.html | 显示**路线类型**界面 |
| 路线详情 | Main.html | 显示**路线详情**界面 |
| 购物车 | Shopping.html | 显示**购物车**界面 |
| 订单填写 | The order.html | 显示**订单填写**界面 |
| 订单支付 | confirm order.html | 显示**订单支付**界面 |

## 业务逻辑数据访问组件

业务逻辑访问组件用来对各类信息的增、删、改、查操作的具体处理执行，相应组件及其相应的具体描述如表 5‑2 所示。

表 5‑2业务逻辑处理单元列表

|  |  |  |
| --- | --- | --- |
| 业务逻辑处理单元 | Js | 说明 |
| 路线类型/详情展示单元 | Line.js | 实现路线类型/详情展示展示 |
| 用户/管理员单元 | People.js | 实现用户/管理员登录 |
| 购物车/订单/团购功能单元 | Product.js | 实现与产品相关功能 |
| 管理员管理单元 | Manage.js | 实现后台所有管理 |

控制层来自表现层的所有主要接口单元都是从Sever.js中进行传值，然后将用户点击需要的值由每一个负责不同域主控制单元进行分值传递。作为一个数据导向的模块存在。如图 5‑1所示

图 ‑1Sever.js部分内容

