

**2019-2020-3学期**

**《信息系统开发综合实训》**

**信息系统设计实训报告**

姓名：张晗天

学号：1708020107

班级：信息管理与信息系统1701班

**2020.7**

目录

[第一章 项目概况及要求 4](#_Toc45121324)

[1.1 编写目的 4](#_Toc45121325)

[1.2 项目背景 4](#_Toc45121326)

[1.3 定义 4](#_Toc45121327)

[1.4 参考资料 4](#_Toc45121328)

[第二章 系统需求分析 5](#_Toc45121329)

[2.1 用户功能要求 5](#_Toc45121330)

[2.2 系统性能要求 5](#_Toc45121331)

[2.2.1 数据准确性 5](#_Toc45121332)

[2.2.2 时间特性 5](#_Toc45121333)

[2.2.3 适应性 6](#_Toc45121334)

[2.3 输入输出要求 6](#_Toc45121335)

[2.4 用户其他要求 6](#_Toc45121336)

[2.4.1 数据管理要求 6](#_Toc45121337)

[2.4.2 故障处理要求 6](#_Toc45121338)

[2.5 可行性论证 7](#_Toc45121339)

[2.5.1 经济可行性评价 7](#_Toc45121340)

[2.5.2 社会因素可行性评价 8](#_Toc45121341)

[2.5.3 技术可行性评价 8](#_Toc45121342)

[第三章 系统分析 9](#_Toc45121343)

[3.1 组织结构及业务流程分析 9](#_Toc45121344)

[3.2 用例图分析 11](#_Toc45121345)

[3.3 实体类图分析 12](#_Toc45121346)

[第四章 系统设计 13](#_Toc45121347)

[4.1 总体设计（功能树设计） 13](#_Toc45121348)

[4.2 顺序图设计 13](#_Toc45121349)

[4.2.1 创建订单信息 13](#_Toc45121350)

[4.2.2 创建线路信息 13](#_Toc45121351)

[4.2.3 销售人员创建并审核预订信息 14](#_Toc45121352)

[4.3 类图设计 14](#_Toc45121353)

[4.3.1 边界类图 14](#_Toc45121354)

[4.3.2 控制类图 15](#_Toc45121355)

[4.3.3 实体类图 15](#_Toc45121356)

[4.4 编码设计（编码规则） 16](#_Toc45121357)

[4.5 模块算法设计 17](#_Toc45121358)

[第五章 系统测试与部署 18](#_Toc45121359)

[5.1 系统架构选择 18](#_Toc45121360)

[5.2 系统部分代码示例 18](#_Toc45121361)

[5.2.1 数据库创建 18](#_Toc45121362)

[5.2.2 Jsp网页（以产品列表为例） 19](#_Toc45121363)

[5.2.3 Java后端处理（以产品列表为例） 20](#_Toc45121364)

[5.3 系统界面实现 21](#_Toc45121365)

[5.4 系统测试 25](#_Toc45121366)

[5.5 系统实施应用 26](#_Toc45121367)

[第六章 其他说明 27](#_Toc45121368)

[第七章 反思日记 28](#_Toc45121369)

# 第一章 项目概况及要求

## 编写目的

在当前的市场环境下，信息爆炸式增长，而作为一个旅行社，需要掌握的信息尤为巨大。在这种情况下，信息系统发挥着越来越重要的作用。为了能够更有效地管理企业，降低人力物力的开支成本。需要为旅行社企业开发信息系统。经过对项目进行的详细调查，本文从技术上、经济上、管理上进行开发的可行性分析并写出该分析报告。

## 项目背景

我国的旅行社目前正处于一个科学技术日新月异飞速向前发展的环境中。信息技术和通信技术以令人目不暇接的速度发展，尤其是互连网络的广泛流行，使得各种服务信息已近乎透明，且个性突出的游客们已经不再满足于死板化的标准化的旅游项目，要求量身定做。

旅游行业内的竞争日趋激烈，这就要求旅行社必须采取现代化的管理手段，利用计算机的发展成果结合旅行社的业务特点，建立高效的旅行社信息系统，以迅速、准确地捕捉客户需求，并加以高效回应，使游客满意，才能摆脱被时代寄出的危险，并在业内脱颖而出。

## 定义

旅行社信息系统是对旅行社经营的所有信息进行综合管理和控制的以人为主体的人机系统。它的职能是对旅行社生产服务过程的管理实现信息化，从而提高旅行社的生产率和管理效率，同时提高旅行社的市场竞争能力，满足当代民众旅游的个性化服务要求。

## 参考资料

《旅游管理信息系统》 查良松、陆均良、罗仕伟著：高等教育出版社

# 第二章 系统需求分析

## 用户功能要求

需要满足的功能包括但不限于：团队业务管理系统、旅游路线管理、团队人员管理、财务管理、系统维护、系统帮助。

本系统的主要功能包括：发布旅行线路，接受订单，预付结算，订单管理，退单、订单完成反馈、旅客数据管理、导游数据管理、酒店数据管理。

## 系统性能要求

为了满足旅行社管理的要求，该系统需要对大量的信息做出最快的处理，所以需要很大的主存容量，以及强大的数据库支持。

### 数据准确性

“十三五”期间，我国旅游业发展迅猛，产业规模持续扩大，产品体系日益完善，市场秩序不断优化，旅游及相关产业逐渐成为国民经济新的增长点。国内游市场是全国旅游市场的主力军，无论从人数还是贡献的旅游收入来看，都是旅游业主要的增长点。随着我国居民收入水平的不断上升，旅游景区建设质量的提高、旅游内涵的不断丰富，国内旅游规模持续快速增长，这就使得数据的准确性变得特别重要。

### 时间特性

一个旅游产品从开发出来投放市场到最后被淘汰退出市场的整个过程，一条旅游路线、一个旅游活动项目、一个旅游景点、一个旅游地开发大多都将遵循一个从无到有、由弱至强、然后衰退、消失的时间过程。对于各种各样的旅游产品，也有不同的生命周期。

### 适应性

人们对旅游的需求也发生着变化。游客的结构也在发生着巨大的变化，人们对旅游商品的需求也在随之发生着变化，旅行社的信息系统也必须有新的标准。很多信息系统的理念已经滞后于旅游发展的潮流，新开发的系统必须适应当下。

## 输入输出要求

输出内容包括但不限于团员基本旅游信息及旅游线路的基本情况、导游的基本信息、导游与景点分布情况、景点与游客的分布情况。

输入内容包括但不限于团员的基本信息、旅游路线的基本情况、导游的基本信息、景点与导游的分布。

## 用户其他要求

### 数据管理要求

1) 各类数据必须是有依据的，能够辨认数据的内容、用途和使用方法；

2) 必须经过合法的手续和规定的渠道采集、加工、处理和传播数据；

3) 数据应只用于明确规定的目的，未经批准不得它用；采用的数据范围应与规定用途相符。

### 故障处理要求

1) 信息系统出现无法进行本地解决的，应向上级领导及上级部门进行申告故障；

2) 如果故障问题比较严重并牵扯到相关部门，在解决故障期间应给相关部门进行通知，提前做好备份工作；

3) 厂商技术人员现场处理故障时，当地维护人员应全程陪同并积极协助，并在故障解决后进行书面确认。

## 可行性论证

该信息系统的目标可以分为以下几点：第一，紧密地将客户、部门和同行联系起来，快速的掌握新的旅游信息，增加顾客消费欲望；第二，对业务流程进行规范化管理，提高工作效率并定时检查漏洞；第三，对所有的客户信息进行记录储存，为下次服务提供便利；第四及时获取交通、酒店信息，减少繁杂多余的浪费。

### 经济可行性评价

#### 支出

支出费用包括但不限于：软件费用，硬件费用，专业人员费用，其他各种费用。

#### 支出

收益包括但不限于：

1) 直接收入：员工人员精简，效率提高；客户流量增加；减少手工造成的各种不必要费用。

2) 间接费用：提高了公司的管理水平；信息化的成功会带动公司声誉的提高。

#### 敏感性分析

系统的本质是加强信息管理，提高业务处理能力。再项目合同签订时，我们会将各种风险及相应的金额记录在案，若系统应用过程中配置发生了变化，我们会派技术人员进行相应的沟通和维护。

### 社会因素可行性评价

#### 法律方面可行性

对于法律方面的分析，主要有下面几点：

1) 所有软件都使用正版。

2) 所有技术资料都由提出方管理。

3) 合同进行规定，确定违约责任。

#### 使用方面可行性

针对目前旅游管理方面所暴露出来的问题，如旅游信息流通不畅、价格信息不透明等，都缺乏一套切实可行的旅游产品与服务信息系统，来取代传统的手工决策方式。开发这样一套旅游管理信息系统，可降低工作人员的工作强度，提高效率，会得到企业上下员工的一致同意，充分利用企业现有资源。

### 技术可行性评价

根据以上的系统功能分析，从技术角度分析，这项开发所涉及的专业技术如下：

* 技术平台 / 操作系统：Windows 10操作系统。
* 开发工具：IntelliJ IDEA Community Edition，Visual Studio等。
* 数据库管理系统：MySQL。

采用现代化且统一的计算机信息网络系统，能够有效帮助旅游企业优化旅游行业，使其在行业内处于优势地位，并相对长期地保持这种战略优势。

该系统是在国内经验的基础上，依托互联网为旅游产业信息化管理系统提供的高效服务，对旅游行业的规范化管理器到了极大的推动作用。旅行社可依托这一平台加强管理和监控。并且，当前的科学技术能够支持系统的运作，性能较为完善，维护性很强，风险较低。

# 第三章 系统分析

## 组织结构及业务流程分析

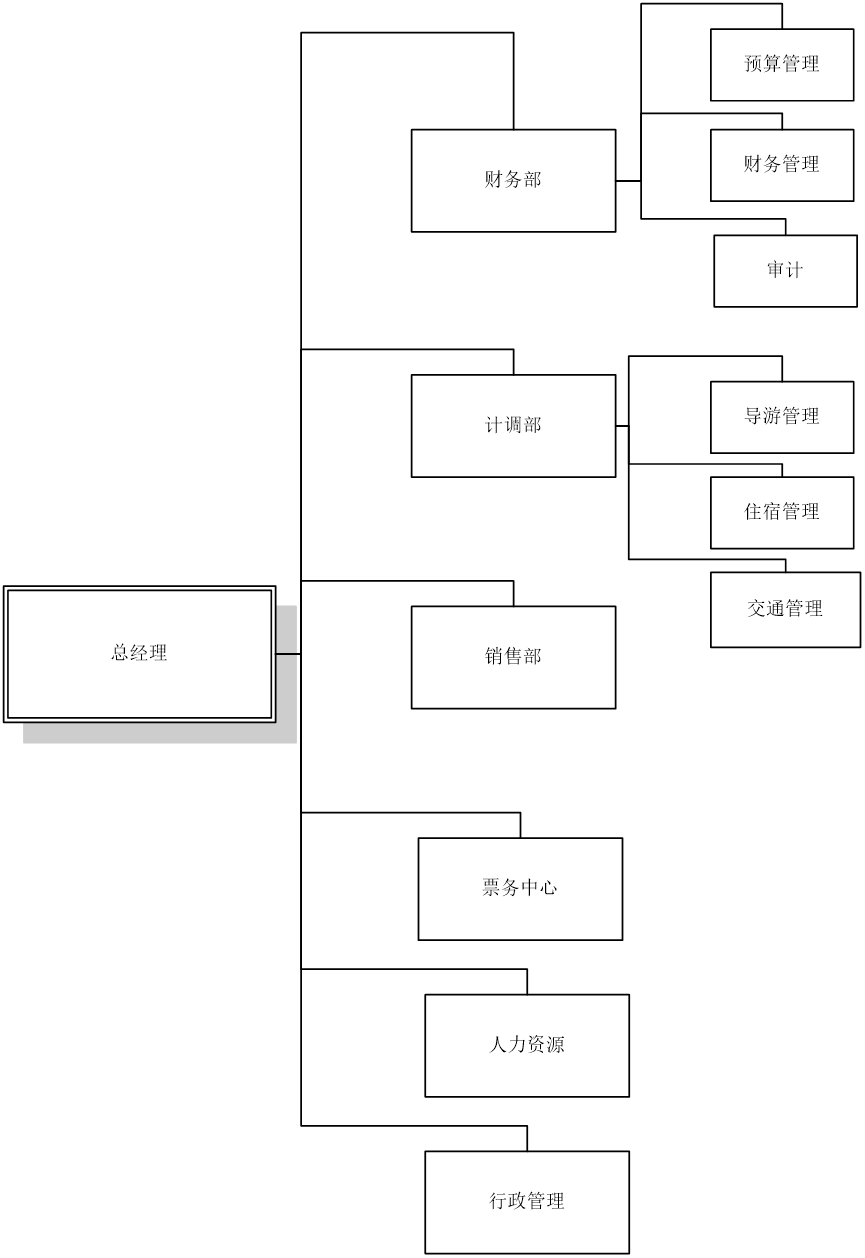


图3-1 组织结构图

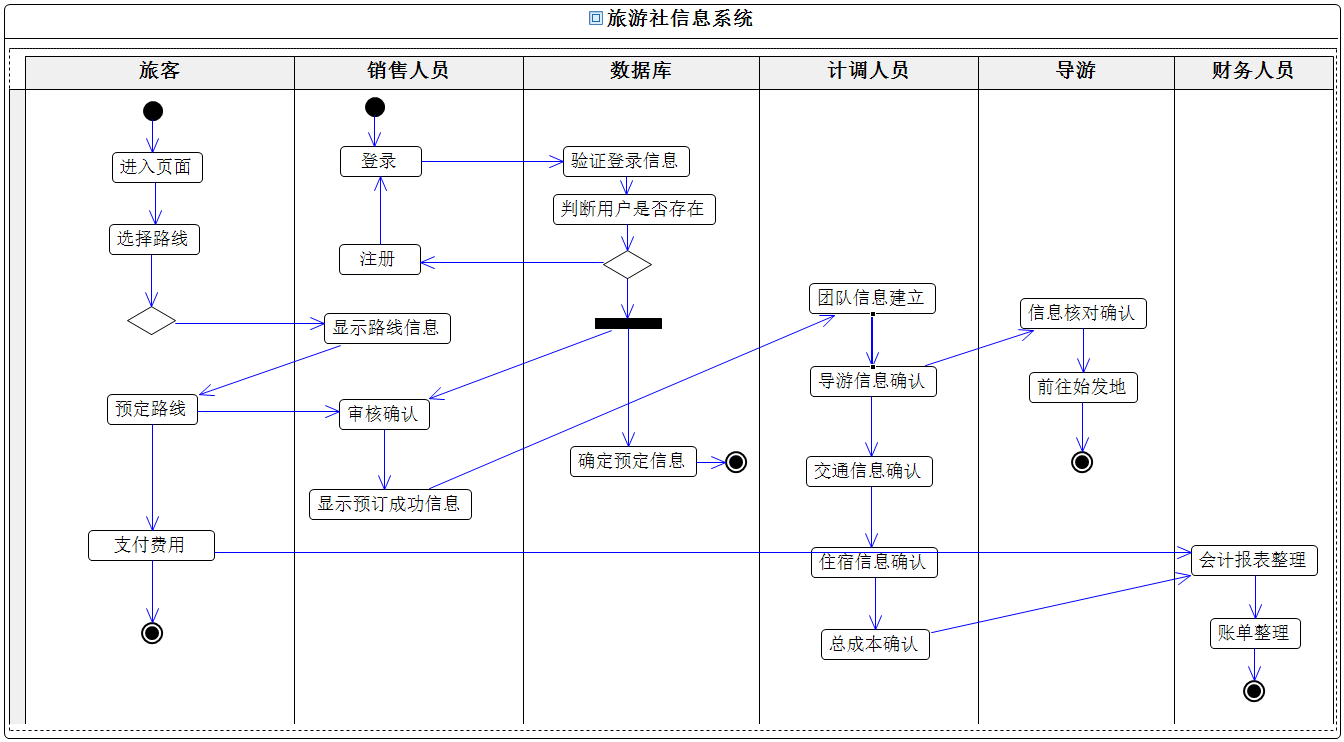


图3-2 业务流程分析图

以散客组团线路销售为例。首先由游客个人产生需求，对旅行社内部产生新的订单。销售部会根据用户的需求进行推荐，用户也可以进行线路信息查看之后下达预订。销售部会接受用户下达的新订单，并且确定价格和预订订单相关信息计划书。之后，组团部对单项订单进行处理，组成团队，并将预定信息产生的订单发给财务部进行留档和处理。组成团队之后，向计调部发送信息，计调部对各项产品和服务进行确认和反馈。完成安排之后，向票务中心发送请求，进行票务的购买。最后，财务部进行核算，留表存档。

## 用例图分析

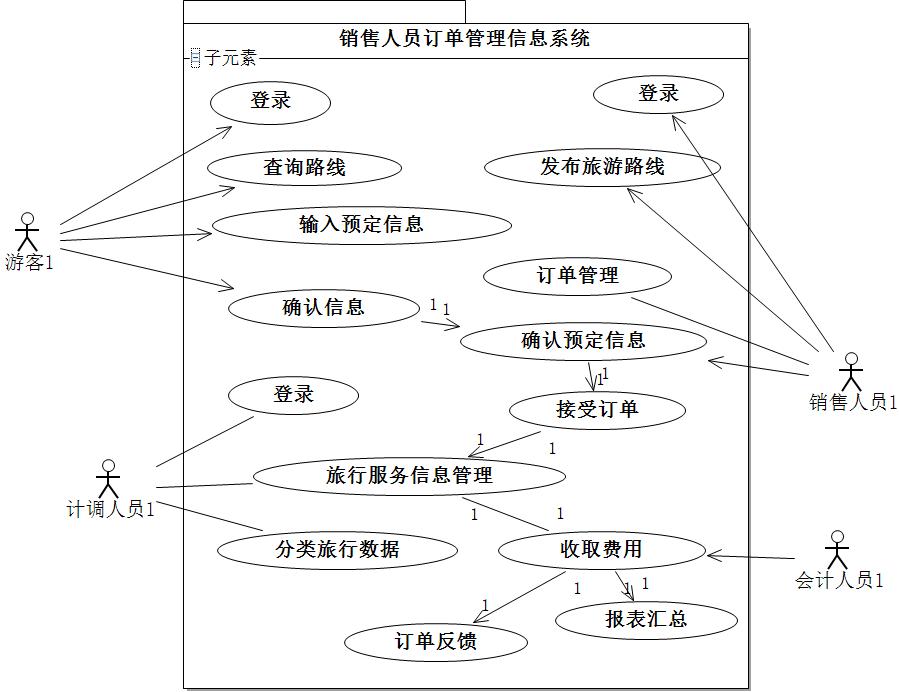


图3-3 订单管理用例分析图

|  |  |  |
| --- | --- | --- |
| **用例名称** | 旅行社信息系统-销售人员订单管理信息系统 | |
| **简要说明** | 完成旅行社订单交易中间环节的信息流动和存储登记 | |
| **参与者** | 顾客、销售人员、计调人员、财务人员 | |
| **前置条件** | 用户或管理员成功登陆系统 | |
| **后置条件** | 系统成功存储相关操作数据和内容 | |
| **基本事件流** | 用户 | 系统 |
| 1. 用户输入账号密码登录 | 1.1 系统显示用户的登录信息 |
| 1. 用户发起查看线路信息的请求 | 2.1 系统显示相关的线路信息 |
| 1. 用户发起下单请求 | 3.1 系统显示下单成功消息 |
| 1. 销售人员对线路信息进行增删改查 | 4.1 系统显示线路数据信息 |
| 1. 销售人员对订单进行审核确定 | 5.1 系统显示未审核订单信息 |
| 1. 计调人员管理订单服务 | 6.1 系统显示组团信息 |
| 1. 财务人员进行会计分析 | 7.1 系统显示相应订单的成本与销售收入 |
| 1. 财务人员进行财务报表分析 | 8.1 系统显示财务报表 |
| **异常事件流** | 1.1 用户登陆失败，弹出验证错误提示 | |
| 2.1 系统显示下单失败，弹出错误提示 | |

表3-1 订单管理用例分析表

## 实体类图分析

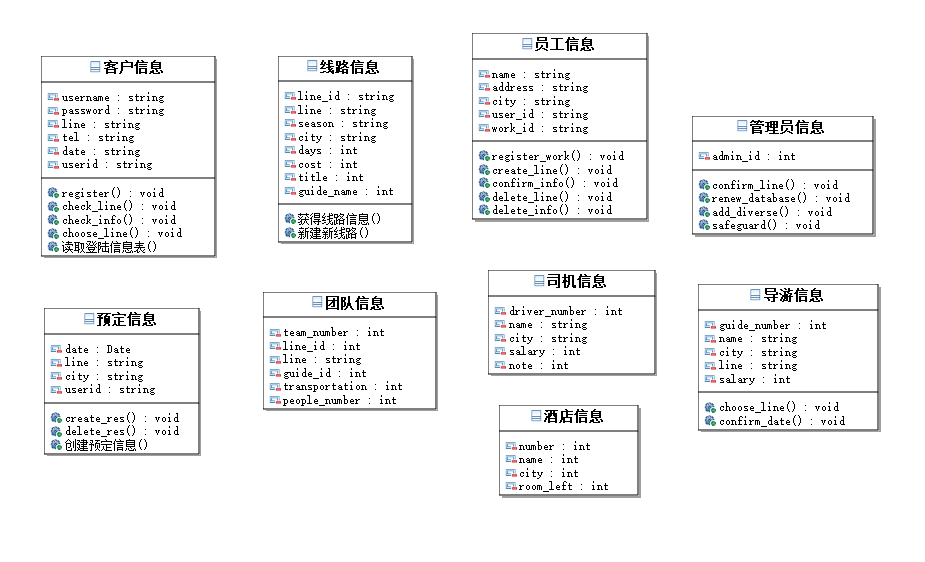


图3-4 实体类图

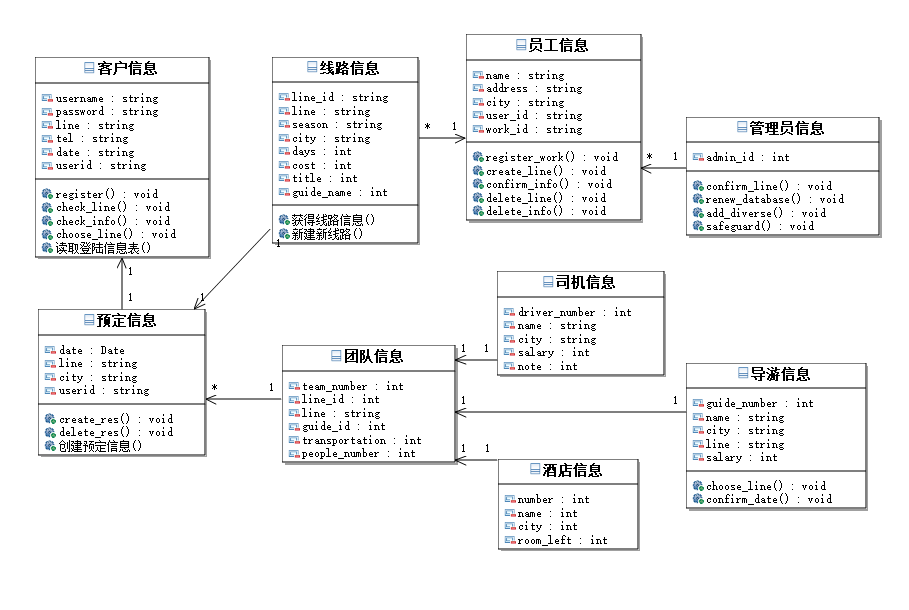


图3-5 实体类联系图

# 第四章 系统设计

## 总体设计（功能树设计）

发布旅行线路，接受订单，预付结算，订单管理，退单、订单完成反馈。

## 顺序图设计

### 创建订单信息

在创建预定信息之前，需要先登录，并且在线路信息查询板块确定需要预定的线路信息，然后进行预定信息的创建。顺序图如下所示：

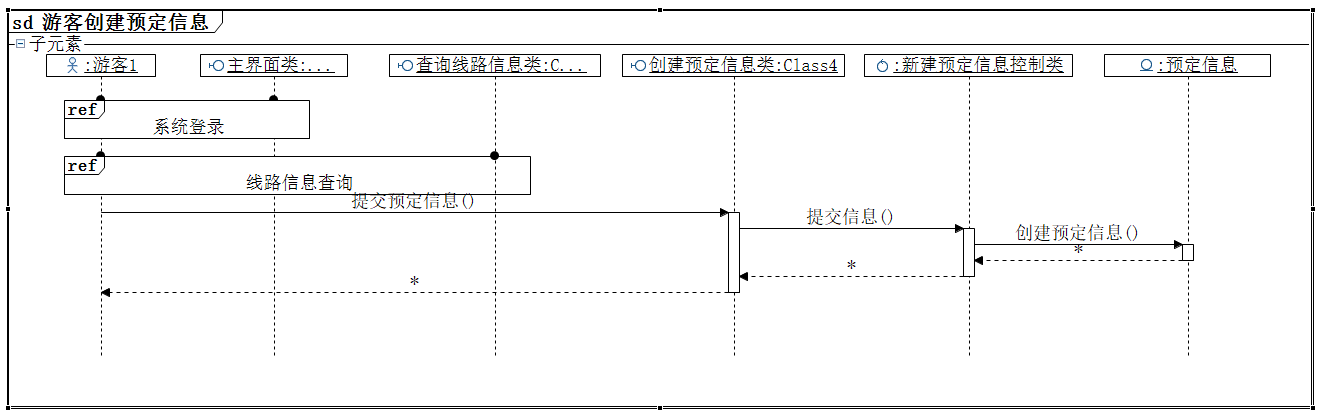


图4-1 创建订单信息顺序图

### 创建线路信息

销售人员创建线路信息前，需要先登录，然后再创建线路信息类提交请求。顺序图如下：

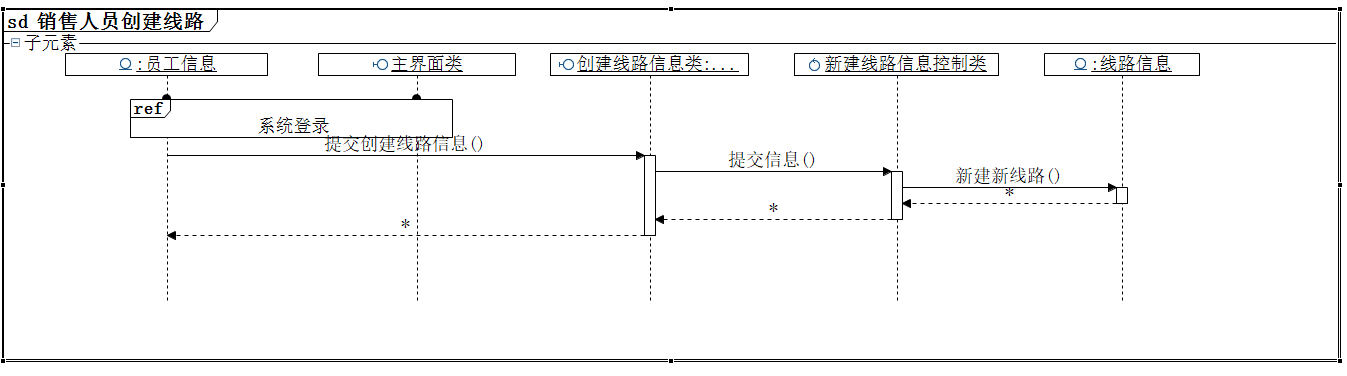


图4-2 创建线路信息顺序图

### 销售人员创建并审核预订信息

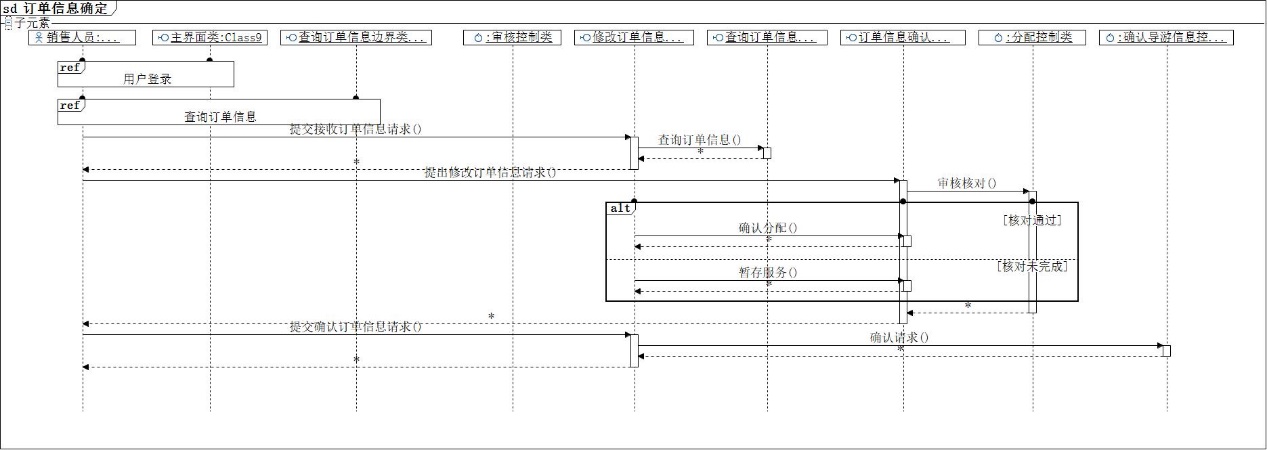


图4-3 审核预定信息顺序图

## 类图设计

### 边界类图

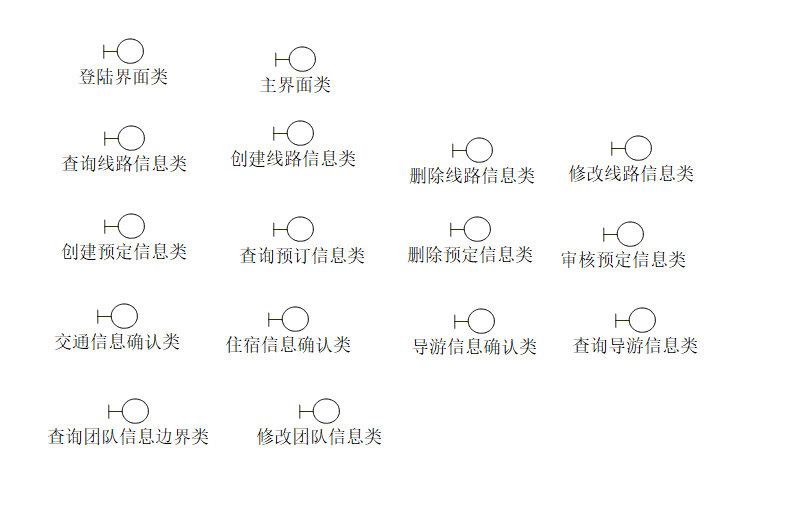


图4-4 边界类图

### 控制类图



图4-5 控制类图

### 实体类图



图4-6 实体类图

## 编码设计（编码规则）

编码名称：乘客编号 数据类型：char 长度：15

编码格式：XXXXXX XXXXXX XXX

注册日期 身份证后六位 名字的大写首字母

编码名称：旅行团编号 数据类型：char 长度：17

编码格式：XXXXXX XXXXXXXX XXX

出发日期 旅游产品编号 出发班次号

编码名称：旅游产品编号 数据类型：char 长度：8

编码格式：X XXXXXX XX

是否为出境 主要目的地 编号

编码名称：订单编号 数据类型：char 长度：30

编码格式：XXXXXX XXXXXXXX XXXXXXXXXXXXXXX X

出发日期 旅游产品编号 客户编号 验证码

## 模块算法设计



图4-7 模块算法设计图

对于系统的开始，销售人员需要执行登录操作，并且需要验证管理员身份。登录成功之后，会在散客系统页面收到顾客从系统中发出的订单。此时，销售人员需要进行订单的审核和确认。核实信息，除了确定相关的填写之外，还需要进行电话或邮件联系，需要更改之后再次进行审核。最后再进行订单上传。

# 第五章 系统测试与部署

## 系统架构选择

本系统的前端开发使用了基于bootstrap和jQuery之上的AdminLTE。它是一款开源的模板主题工具，提供了一系列响应的、可重复使用的组件，并内置了多个模板页面；同时自适应多种屏幕分辨率，兼容PC和移动端。通过AdminLTE，可以快速的创建一个响应式的Html5网站。在GitHub上可以获取相应的源代码和帮助信息。

本系统的后端使用的是Java语言，基于Spring环境、Maven工程和MySQL数据库进行开发的信息系统。

## 系统部分代码示例

### 数据库创建

1. **DROP** **TABLE** IF EXISTS `orders`;
2. **CREATE** **TABLE** `orders`  (
3. `id` **int**(32)  NOT NULL AUTO\_INCREMENT,
4. `orderNum` **varchar**(20) **CHARACTER** **SET** utf8 **COLLATE** utf8\_general\_ci NOT NULL,
5. `orderTime` **timestamp**(0) NULL **DEFAULT** NULL,
6. `peopleCount` **int**(11) NULL **DEFAULT** NULL,
7. `orderDesc` **varchar**(500) **CHARACTER** **SET** utf8 **COLLATE** utf8\_general\_ci NULL **DEFAULT** NULL,
8. `payType` **int**(11) NULL **DEFAULT** NULL,
9. `orderStatus` **int**(11) NULL **DEFAULT** NULL,
10. `productId` **int**(32) NULL **DEFAULT** NULL,
11. `memberId` **int**(32) NULL **DEFAULT** NULL,
12. **PRIMARY** **KEY** (`id`) USING BTREE,
13. **INDEX** `fk\_o\_p`(`productId`) USING BTREE,
14. **INDEX** `fk\_o\_m`(`memberId`) USING BTREE,
15. **CONSTRAINT** `fk\_o\_m` **FOREIGN** **KEY** (`memberId`) **REFERENCES** `member` (`id`) **ON** **DELETE** **RESTRICT** **ON** **UPDATE** **RESTRICT**,
16. **CONSTRAINT** `fk\_o\_p` **FOREIGN** **KEY** (`productId`) **REFERENCES** `product` (`id`) **ON** **DELETE** **RESTRICT** **ON** **UPDATE** **RESTRICT**
17. ) ENGINE = InnoDB **CHARACTER** **SET** = utf8 **COLLATE** = utf8\_general\_ci ROW\_FORMAT = **Dynamic**;
19. **INSERT** **INTO** `orders` **VALUES** ('1', '12345', '2018-02-03 00:00:00', 2, '没什么', 0, 1, 1, 1);
20. **INSERT** **INTO** `orders` **VALUES** ('2', '12346', '2020-04-15 11:15:17', 2, '没什么', 0, 1, 1, 1);

### Jsp网页（以产品列表为例）

1. **<table** id="dataList"
2. class="table table-bordered table-striped table-hover dataTable"**>**
3. **<thead>**
4. **<tr>**
5. **<th** class="" style="padding-right: 0px;"**><input**
6. id="selall" type="checkbox" class="icheckbox\_square-blue"**>**
7. **</th>**
8. **<th** class="sorting\_asc"**>**ID**</th>**
9. **<th** class="sorting\_desc"**>**编号**</th>**
10. **<th** class="sorting\_asc sorting\_asc\_disabled"**>**产品名称**</th>**
11. **<th** class="sorting\_desc sorting\_desc\_disabled"**>**出发城市**</th>**
12. **<th** class="sorting"**>**出发时间**</th>**
13. **<th** class="text-center sorting"**>**产品价格**</th>**
14. **<th** class="sorting"**>**产品简述**</th>**
15. **<th** class="text-center sorting"**>**状态**</th>**
16. **<th** class="text-center"**>**操作**</th>**
17. **</tr>**
18. **</thead>**
19. **<tbody>**

22. **<c:forEach** items="${productList}" var="product"**>**
24. **<tr>**
25. **<td><input** name="ids" type="checkbox" value="${product.id}"**></td>**
26. **<td>**${product.id }**</td>**
27. **<td>**${product.productNum }**</td>**
28. **<td>**${product.productName }**</td>**
29. **<td>**${product.cityName }**</td>**
30. **<td>**${product.departureTimeStr }**</td>**
31. **<td** class="text-center"**>**${product.productPrice }**</td>**
32. **<td>**${product.productDesc }**</td>**
33. **<td** class="text-center"**>**${product.productStatusStr }**</td>**
34. **<td** class="text-center"**>**
35. **<button** type="button" class="btn bg-olive btn-xs" onclick="location.href='${pageContext.request.contextPath}/product/findById.do?id=${product.id}'"**>**详情**</button>**
36. **<a** href="${pageContext.request.contextPath}/product/openProduct.do?id=${product.id}" class="btn bg-olive btn-xs" onclick='confirm("你确认要开启此产品吗？")'**>**开启**</a>**
37. **<a** href="${pageContext.request.contextPath}/product/deleteProduct.do?id=${product.id}" class="btn bg-olive btn-xs" onclick='confirm("你确认要删除吗？")'**>**删除**</a>**
38. **</td>**
39. **</tr>**
40. **</c:forEach>**
41. **</tbody>**
42. <!--
43. **<tfoot>**
44. **<tr>**
45. **<th>**Rendering engine**</th>**
46. **<th>**Browser**</th>**
47. **<th>**Platform(s)**</th>**
48. **<th>**Engine version**</th>**
49. **<th>**CSS grade**</th>**
50. **</tr>**
51. **</tfoot>**--**>**
52. **</table>**

### Java后端处理（以产品列表为例）

1. @RequestMapping("/findAll.do")
2. //@RolesAllowed("ADMIN")
3. **public** ModelAndView findAll() **throws** Exception {
4. ModelAndView mv = **new** ModelAndView();
5. List<Product> products = productService.findAll();
6. mv.addObject("productList", products);
7. mv.setViewName("product-list");
8. **return** mv;
9. }
11. @Select("select \* from product")
12. **public** List<Product> findAll() **throws** Exception;
14. @Insert("insert into product values(id,#{productNum},#{productName},#{cityName},#{departureTime},#{productPrice},#{productDesc},#{productStatus},#{guideNum},#{hotelNum},'反馈：',#{peopleTotal},#{timeTotal})")
15. **void** save(Product product) **throws** Exception;
17. @Select("select \* from product where id=#{id}")
18. @Results({
19. @Result(id = **true**, property = "id", column = "id"),
20. @Result(property = "productNum", column = "productNum"),
21. @Result(property = "productName", column = "productName"),
22. @Result(property = "cityName", column = "cityName"),
23. @Result(property = "departureTime", column = "departureTime"),
24. @Result(property = "productPrice", column = "productPrice"),
25. @Result(property = "productDesc", column = "productDesc"),
26. @Result(property = "productStatus", column = "productStatus"),
27. @Result(property = "peopleTotal", column = "peopleTotal"),
28. @Result(property = "timeTotal", column = "timeTotal"),
29. @Result(property = "hotel",column = "hotelNum",javaType = Hotel.**class**,one = @One(select = "com.zht.dao.iHotelDao.findById")),
30. @Result(property = "guide",column = "guideNum",javaType = Guide.**class**,one = @One(select = "com.zht.dao.iGuideDao.findById")),
31. })
32. **public** Product findById(**int** id) **throws** Exception;

## 系统界面实现



图5-1 登录输入

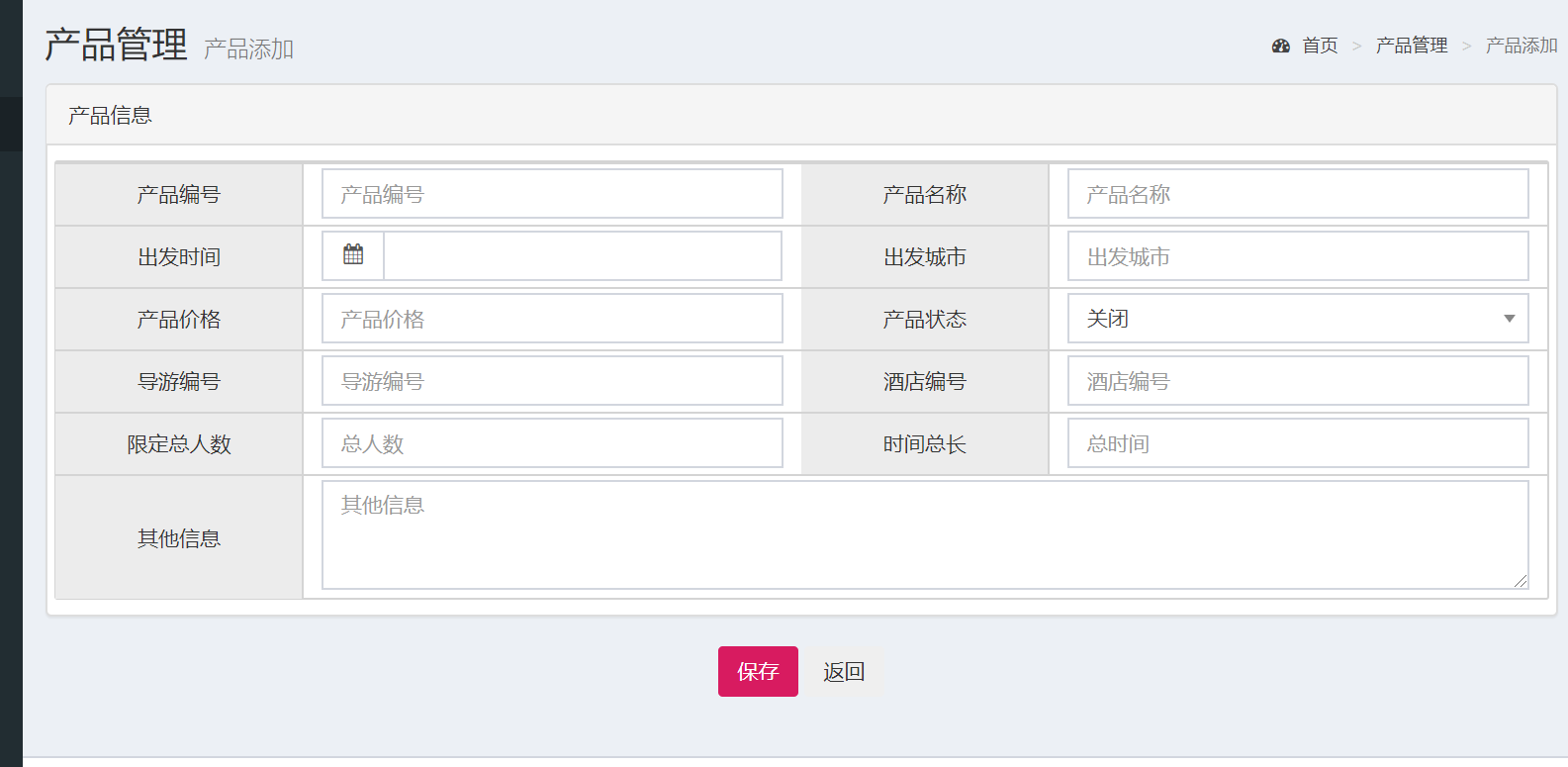


图5-2 产品信息输入

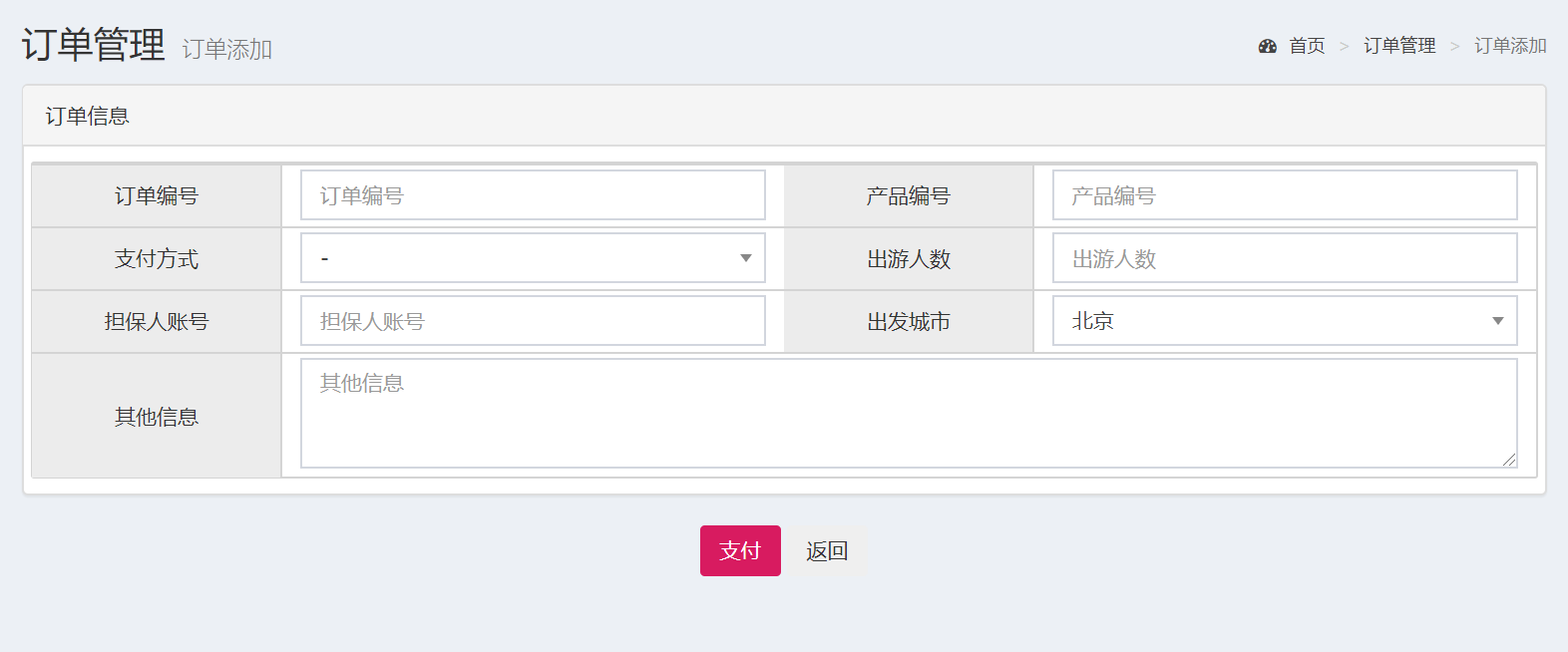


图5-3 订单信息输入



图5-4 旅客信息输入

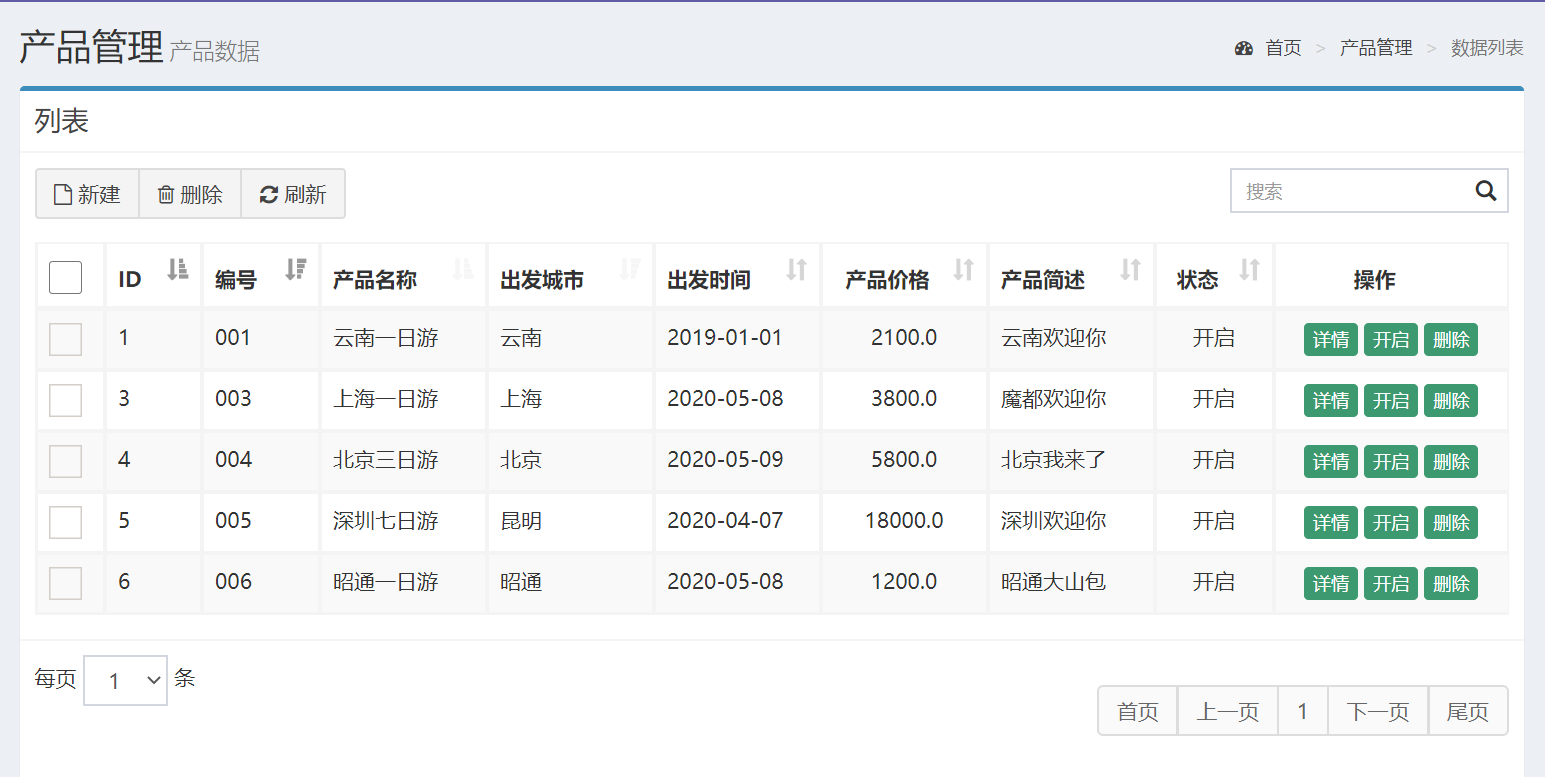


图5-5 产品数据输出图

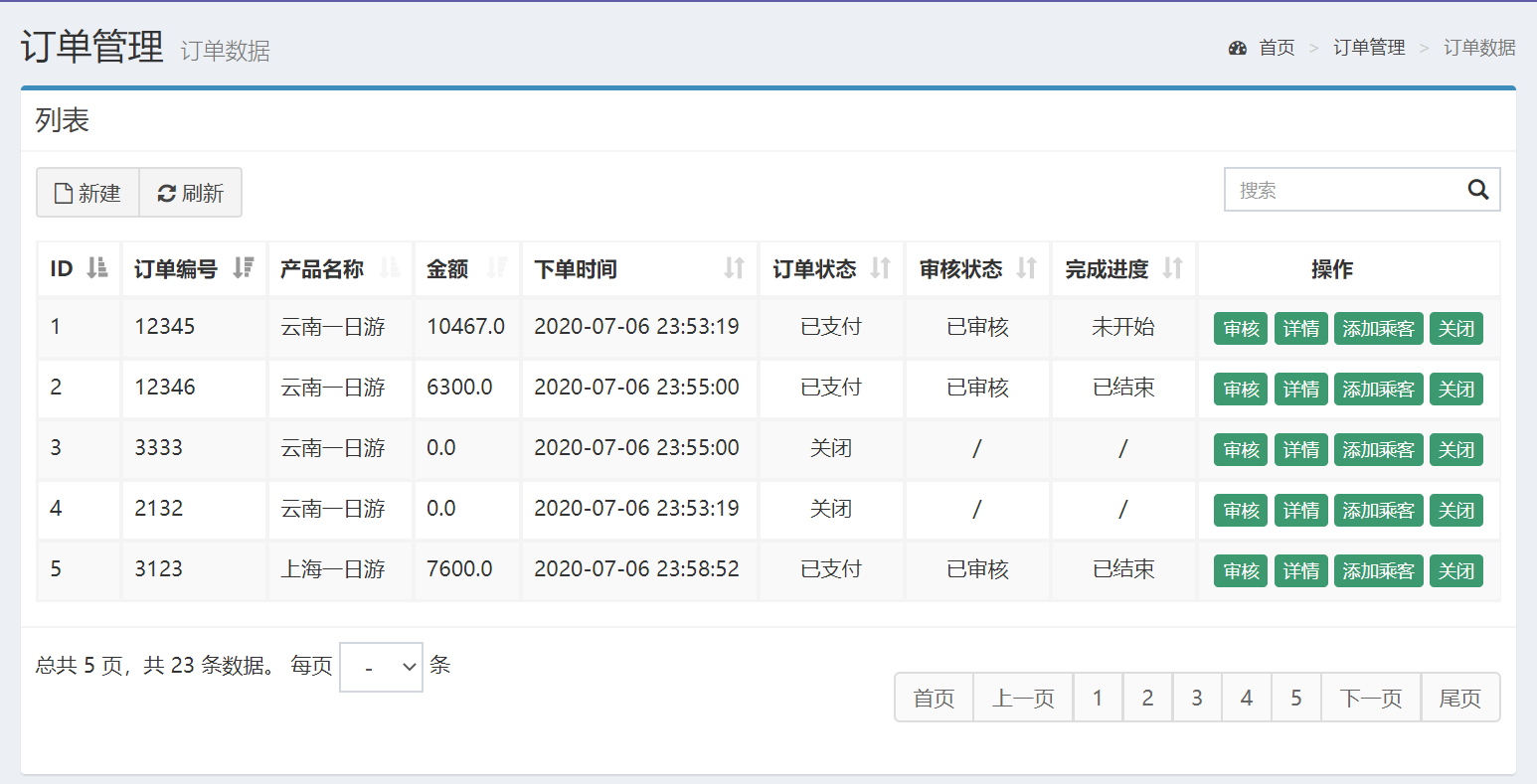


图5-6 订单数据输出图

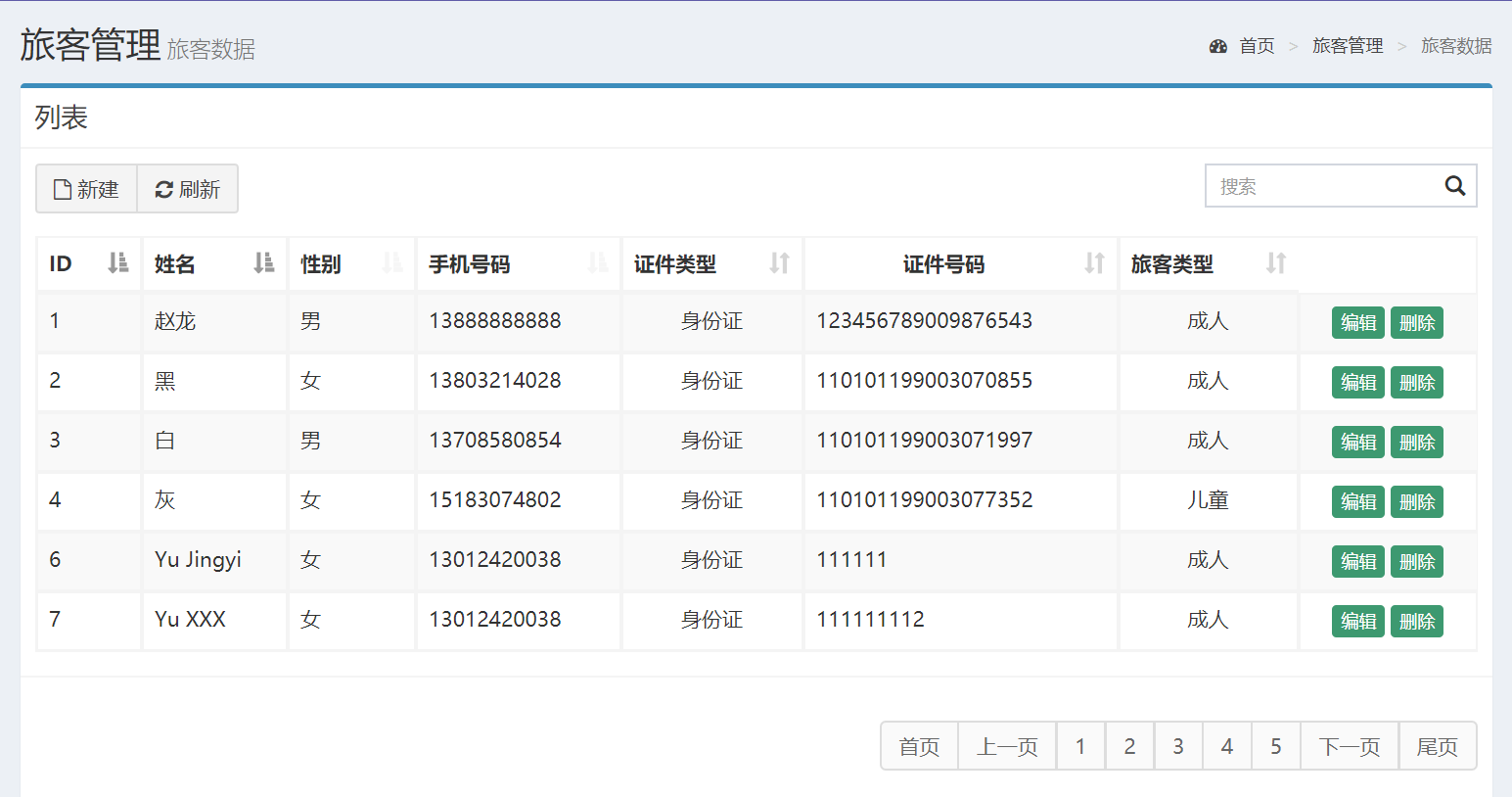


图5-7 旅客数据输出图

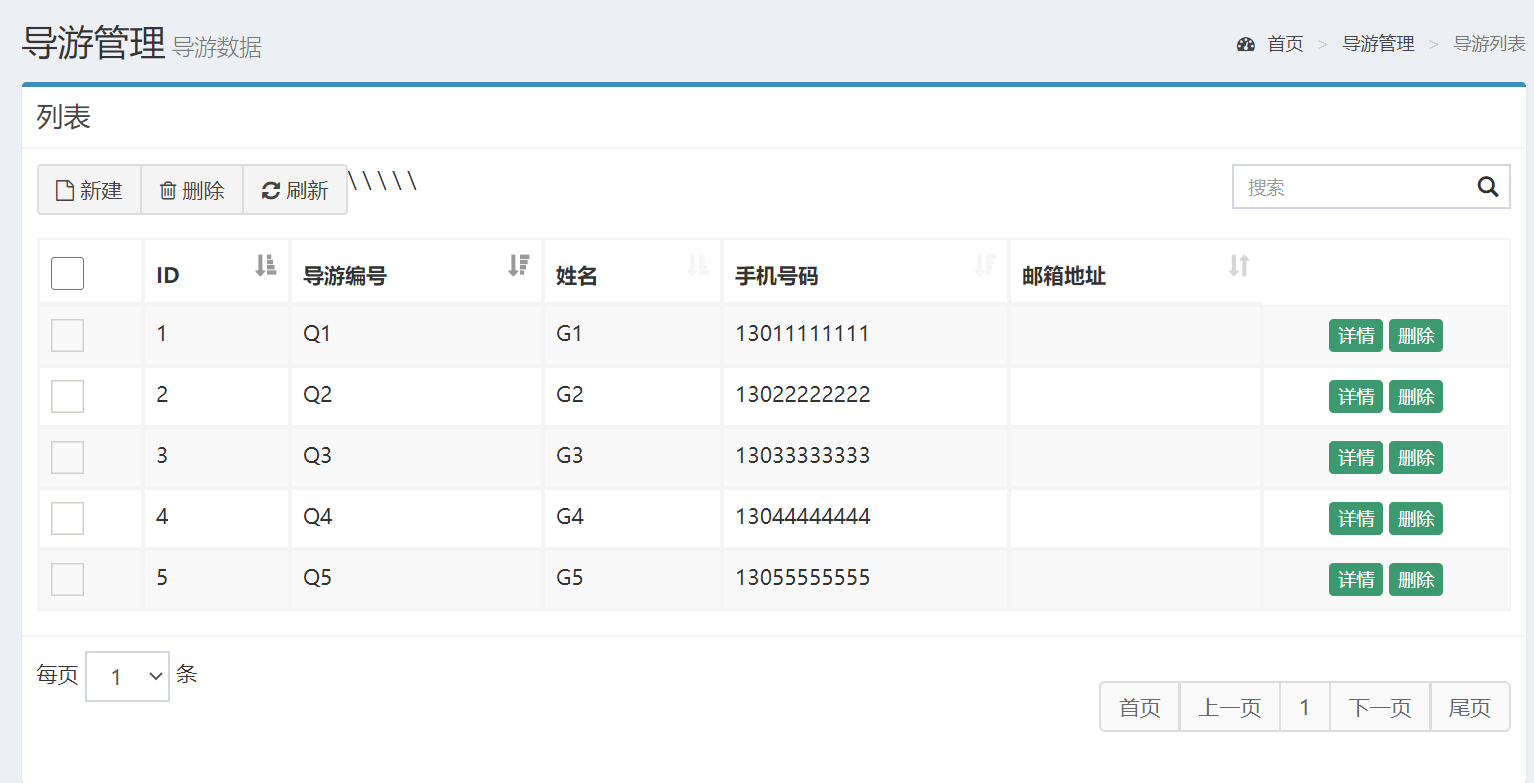


图5-8 导游数据输出图



图5-9 酒店数据输出图

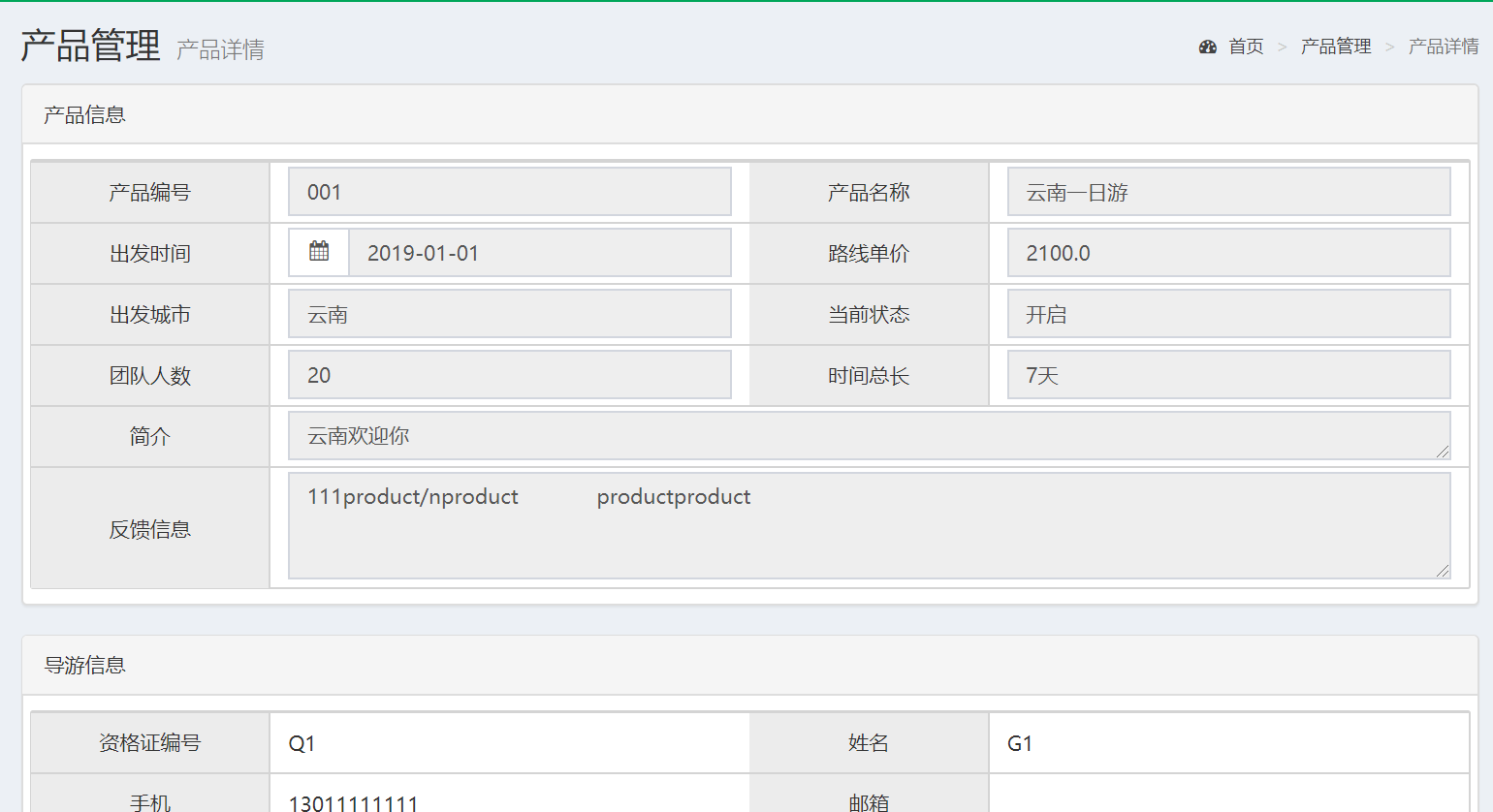


图5-10 产品详情输出图

图5-11 订单详情输出图

## 系统测试

#### 数据格式测试

在数据库中，由于MySQL数据库的特征，必须要设置一个int格式的id值，所以在后端java输入时不能使用int格式传入各种id值，才能使数据库成功记录数据。

1. @Delete("delete from product where id=#{productId}")
2. **void** deleteRoleById(**int** productId) **throws** Exception;

#### 数据库测试

在数据库中，由于后面要进行新外键的生成或修改，需要进入原表中，加入到Indexes选项中，才可以继续使用外键。

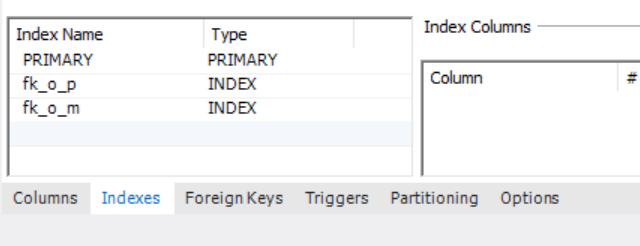


图5-12 订单详情输出图

#### 多表联合查询测试

在前端进行多表联合查询中，后端不能简单的应用类中的数据，需要加上相关的关联代码，才能显示出存储在其他表中的信息

1. @Select("select \* from product where id=#{id}")
2. @Results({
3. @Result(id = **true**, property = "id", column = "id"),
4. @Result(property = "productNum", column = "productNum"),
5. @Result(property = "productName", column = "productName"),
6. @Result(property = "cityName", column = "cityName"),
7. @Result(property = "departureTime", column = "departureTime"),
8. @Result(property = "productPrice", column = "productPrice"),
9. @Result(property = "productDesc", column = "productDesc"),
10. @Result(property = "productStatus", column = "productStatus"),
11. @Result(property = "peopleTotal", column = "peopleTotal"),
12. @Result(property = "timeTotal", column = "timeTotal"),
13. @Result(property = "hotel",column = "hotelNum",javaType = Hotel.**class**,one = @One(select = "com.zht.dao.iHotelDao.findById")),
14. @Result(property = "guide",column = "guideNum",javaType = Guide.**class**,one = @One(select = "com.zht.dao.iGuideDao.findById")),
15. })
16. **public** Product findById(**int** id) **throws** Exception;

## 系统实施应用

在浏览器中输入[localhost:8888/zht\_web/](127.0.0.1:8888/zht_web/)，即可进入系统的登陆界面。系统的登陆需要账号密码，需要在数据库中注册才能进行下一步界面，否则会跳转到失败界面。

# 第六章 其他说明

在系统的开发过程中，借助了大量网上的帮助，经过这几天的时间，完成的功能并不算多，功能也并不完善，有一些功能的开发逻辑并不规范，也容易产生各种功能打架的现象。在今后的学习中，我会继续学习相关知识，在这个领域挖得更深。

# 第七章 反思日记

|  |  |
| --- | --- |
| 日期 | 实训工作内容及反思日志 |
| 6．22 | 工作内容：  听取老师相关指导意见；撰写分析设计报告。  反思日志：  重新翻看之前的报告，发现有一些内容（比如各种类的属性等），写的有些粗糙，并且重新进行更正。 |
| 6.23 | 工作内容：  撰写分析设计报告。  反思日志：  与老师进行商讨之后，选择了适合的功能，进行下一步的开发功能研究和面向对象设计。 |
| 6.24 | 工作内容：  撰写分析设计报告。  反思日志：  有一些类图的设计不够准确，重新下载trufun之后重新进行设计。 |
| 6.28 | 工作内容：  选取合适的工具和系统框架。  反思日志：  因为最近刚刚学完了Android应用的开发，所以对于开发APP有一定的兴趣，但是对于使用的展示界面的美观程度有点悲观。对于小程序来说，是从来没有接触过的开发，而且感觉很难进行数据库的连接。网站的开发设计在于代码量太过庞大，对于目前需要进行设计的网页来说，耗费的时间会比较多，明天再看看网站。 |
| 6.29 | 工作内容：  选取合适的工具和系统框架。  反思日志：  在网上进行搜索之后，通过“后台管理系统”这一关键词的搜索，了解到了网上有很多已经编写好的前端框架，只需要进行一定程度的修改之后就可以运用在后端编写上。对于几种比较热门的网络框架中，最终选择了AdminLTE的开发框架，并进行了IDEA的安装配置和Maven工程的配置。 |
| 6.30 | 工作内容：  数据库和项目的类的创建和控制  反思日志：  在数据库的创建过程中，由于对于外键功能的不熟悉，造成了多次的崩溃。在网上搜索到相关的解决方法之后，及时进行了改正，并且在工程中保存了相应的SQL文件，防止下一次再一次崩溃。  对于每一个类，都进行了相关的创建。并且通过网络的资料可了解到，IDEA中内置了getter and setter的功能，可以自动为该类中的各种属性进行方法的创建，节省了许多时间。 |
| 7.1 | 工作内容：  登陆界面的设计、产品的列表展示和创建的相关功能、网页的设计和创建和后端的设计。  反思日志：  在登陆界面中，我设置了一个解密的工具。由于数据库中存储的是经过加密之后的密码，所以需要进行解密才可以正确进入系统。我在这个环节中没有检查出错误，导致花了很多时间才解决这个问题。  在网页的设计当中，由于一些@表示没有添加，导致没能正确将该class的功能传递进去所以花费了一些时间，在后面的开发中要注意这个问题。 |
| 7.2 | 工作内容：  产品详情展示的撰写和网页设计。  反思日志：  产品详情的展示需要重新设计网页和后端功能。  产品详情的展示由于需要与其它表关联，进行多表关联查询，所以代码需要增加相关的程序。  网页的设计中，需要重新设计，使用了AdminLTE中提供的一些列表框。 |
| 7.3 | 工作内容：  订单列表、订单详情的展示的撰写和网页设计。  反思日志：  订单列表的设计和详情页面的设计与产品类似，需要展示的功能和信息也比较多，所以空间也比较拥挤。因为多选功能太过繁琐，并且会在后续的功能中提供关闭功能，所以取消了多选框这一功能。 |
| 7.4 | 工作内容：  增加翻页和删除功能。  反思日志：  在产品页面提供了多选删除的功能。  在订单页面提供了翻页的功能，并可以显示一页显示多少数据。 |
| 7.5 | 工作内容：  增加产品的相关属性。  反思日志：  为了使产品详情的列表增加一定的可读性，在产品类增加了一些相关属性，这使得SQL语句需要重新书写。 |
| 7.6 | 工作内容：  进行了退单、删除、关闭和反馈等几个功能的创建。  反思日志：  由于几个关闭订单、关闭产品有类似的功能，所以编写的方法也比较类似，但是订单的关闭也会使得审核状态和完成状态变成关闭，所以需要在数据库操作中增加这一项功能。  对于反馈这一功能，通过CONCAT这一数据库操作才能实现，而且由于MYSQL中不能识别出换行符，导致最后的反馈栏会比较拥挤，不够美观，需要填写号码进行分隔。 |
| 7.7 | 工作内容：  进行酒店类和导游类的列表查询功能设计和网页设计，并且对网页进行了一定的设计  反思日志：  酒店和导游两个列的列表的创建比较简单，与之前的三个类的列表结构类似。同样也在@Service这个标记这里绊了一跤。  网页也对颜色和一些小图标进行了重新的规划。AdminLTE中提供了很多可以直接使用的小图标，非常方便。 |
| 7.8 | 工作内容：  进行实训报告的撰写和总体系统的测试。  反思日志：  在最后一天的开发中，我对于系统的所有功能进行了测试，确定无误之后填入了相关的数据，以用于展示。  根据整个系统的功能设计，重新撰写了分析设计报告，并且进行了总结。 |

|  |  |
| --- | --- |
| 总结 | 这是我第一次独立完成一个系统项目，我学习到了很多很多知识和经验。   * 对于系统的开发，进行完备的设计和分析是非常重要的，否则会在后面的开发之中不停修修补补，导致效率变慢。 * 对于网页来说，前端的设计比较费劲，使用一些比较成熟的前端框架是很合适的。 * 对于我来说，已经很久没有真正接触过代码了，Java的代码更是很久没写过了，所以一开始比较生疏，但马上就找回了以前的感觉。   通过这一次系统的搭建，我学习到了很多，会带着这一份经验，运用到未来的学习和工作生活当中去。 |