目录

[更新历史 2](#_Toc465533473)

[一、 引言 4](#_Toc465533474)

[(一) 编制目的 4](#_Toc465533475)

[(二) 词汇表 4](#_Toc465533476)

[(三) 参考资料 5](#_Toc465533477)

[二、 产品概述 6](#_Toc465533478)

[三、 体系结构设计概述 7](#_Toc465533479)

[四、 结构视角 9](#_Toc465533480)

[(一) 展示层的分解 9](#_Toc465533481)

[1. 用户界面层模块的职责 10](#_Toc465533482)

[2. 用户界面模块的接口规范 10](#_Toc465533483)

[3. 用户界面层模块的接口规范 10](#_Toc465533484)

[4. 用户界面模块设计原理 11](#_Toc465533485)

[(二) 业务逻辑层的分解 11](#_Toc465533486)

[1. logIn模块 11](#_Toc465533487)

[(三) 数据层的分解 32](#_Toc465533488)

[1. 数据层模块的职责 32](#_Toc465533489)

[2. 数据层模块的接口规范 33](#_Toc465533490)

[五、 信息视角 43](#_Toc465533491)

[(一) 数据持久化对象 43](#_Toc465533492)

[(二) 文件格式 44](#_Toc465533493)

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **修改日期** | **修改原因** | **版本号** |
| 冯俊杰 | 2016/10/4 | 最初草稿 |  |
| 全体成员 | 2016/10/10 | 新增逻辑分包 |  |
| 全体成员 | 2016/10/11 | 新增物理分包 |  |
| 冯俊杰 | 2016/10/12 | 修改物理分包 |  |
| 龚尘淼 | 2016/10/12 | 修改逻辑分包 |  |
| 龚尘淼 | 2016/10/12 | 新增promotion、hotel接口规范 |  |
| 董金玉 | 2016/10/12 | 新增member、user接口规范 |  |
| 冯俊杰 | 2016/10/12 | 新增logIn、credit、market接口规范 |  |
| 高源 | 2016/10/12 | 新增order接口规范 |  |
| 董金玉 | 2016/10/13 | member、user接口规范评审后第一次更新 |  |
| 冯俊杰 | 2016/10/13 | logIn、credit、market接口规范评审后第一次更新 |  |
| 高源 | 2016/10/13 | order接口规范评审后第一次更新 |  |
| 冯俊杰 | 2016/10/13 | credit、market接口规范评审后第二次更新 |  |
| 冯俊杰 | 2016/10/13 | 修改物理分包逻辑层依赖关系 |  |
| 龚尘淼 | 2016/10/14 | 修改逻辑图和用户界面跳转图 |  |
| 龚尘淼 | 2016/10/14 | promotion、hotel接口规范评审后第一次更新 |  |
| 冯俊杰 | 2016/10/14 | 修改逻辑分包 |  |
| 高源 | 2016/10/14 | 新增进程图 |  |
| 龚尘淼 | 2016/10/14 | promotion、hotel接口规范评审后第二次更新 |  |
| 董金玉 | 2016/10/14 | member、user接口规范评审后第二次更新 |  |
| 高源 | 2016/10/15 | order接口规范评审后第二次更新 |  |
| 冯俊杰 | 2016/10/15 | 整合、合并文档 | V0.1 |
| 龚尘淼 | 2016/10/14 | promotion、hotel接口规范评审后第三次更新 | V0.2 |
| 高源 | 2016/10/15 | 新增数据持久化对象PO | V0.3 |
| 高源 | 2016/10/16 | 修改更新数据持久化对象PO | V0.3.1 |
| 全体人员 | 2016/10/16 | 修改细节 | V0.4 |
| 高源 | 2016/10/16 | 排版 | V1.0 |
| 全体人员 | 2016/10/25 | 修改VO、PO | V1.1 |

# 引言

## 编制目的

本报告详细完成对互联网酒店预订系统的详细设计，达到指导后续软件构造的目的，同时实现和测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户编写，是了解系统的导航。

## 词汇表

|  |  |
| --- | --- |
| **缩写或单词** | **解释** |
| HRS | 互联网酒店预订系统 |
| order | 订单 |
| hotel | 酒店 |
| guest | 客户 |
| hotelWorker | 酒店工作人员 |
| webMarketer | 网站营销人员 |
| webManager | 网站管理人员 |
| signUp | 注册 |
| logIn | 登录 |
| UI | 界面 |
| BL | 逻辑 |
| Data | 数据 |
| promotion | 促销策略 |
| credit | 信用值 |
| member | 会员 |
| market | 营销 |
|  |  |
|  |  |
|  |  |
|  |  |

## 参考资料

1. 互联网酒店预订系统用例文档
2. 互联网酒店预订系统需求规格说明文档
3. 互联网酒店预订系统体系结构设计文档
4. 丁二玉，刘钦.计算与软件工程（卷二）[M]机械工业出版2012：134—182
5. IEEE std 1471-2000

# 产品概述

参考《互联网酒店预订系统用例文档》和《互联网酒店预订系统需求规格说明文档》中对产品的概括描述。互联网酒店预订系统主要功能见用例图如下。



# 体系结构设计概述

互联网酒店预订系统中，选择了分层体系结构的风格，将系统分为3层（展示层、业务逻辑层、数据层）能够很好的示意整个高层抽象。展示层包括GUI页面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如图1和图2所示。



**图1：分层结构**



**图2：逻辑设计方案**

# 结构视角

## 展示层的分解

根据需求，系统存在38个用户界面（加粗为主界面）： **登录界面**、**注册界面**、**客户主界面**、个人基本信息界面、个人基本信息维护界面、会员注册界面、未评价订单界面、订单评价界面、个人订单概况列表界面、个人订单详情界面、个人订单撤销界面、商圈地址选定界面、酒店概况列表界面、订单生成界面、酒店搜索界面、酒店详情界面、会员注册界面、**酒店工作人员界面**、酒店信息界面、酒店信息维护界面、客房信息维护界面、酒店促销策略界面、订单列表界面、订单详情界面、剩余客房信息界面、线下客户入住和退房处理界面、**网站营销人员界面**、用户编号录入界面、信用充值界面、异常订单概况列表界面、异常订单详情界面、异常订单撤销界面、网站促销策略界面、会员等级制定界面、**网站管理人员界面**、会员信息界面、会员信息维护界面、酒店信息添加界面。

界面跳转如下图所示。



### 用户界面层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| MainFrame | 界面frame，负责界面的显示和界面的跳转 |

### 用户界面模块的接口规范

### 用户界面层模块的接口规范

|  |  |  |
| --- | --- | --- |
| mainUI | 语法 | init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示Frame |
| logInUI | 语法 | init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示LoginPanel |
| guestUI | 语法 | init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示guestPanel |
| hotelWorkerUI | 语法 | init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示hotelWorkerPanel |
| webMarketerUI | 语法 | init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示webMarketerPanel |
| webManagerUI | 语法 | init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示webManagerPanel |

1. **用户界面层模块需要的服务接口**

|  |  |
| --- | --- |
| 服务名 | 服务 |
| businessLogicService.logInBLService | 登录界面和客户注册的业务逻辑接口 |
| businessLogicService.orderBLService | 负责处理与订单相关的接口 |
| businessLogicService.hotelBLService | 负责处理与酒店相关的接口 |
| businessLogicService.promotionBLService | 负责处理与促销策略相关的接口 |
| businessLogicService.creditBLService | 负责处理与信用值相关的接口 |
| businessLogicService.memberBLService | 负责处理与会员相关的接口 |
| businessLogicService.userBLService | 负责处理与所有用户信息相关的接口 |
| businessLogicService.marketBLService | 负责处理与会员等级制度信息相关的接口 |

### 用户界面模块设计原理

用户界面拟用java的Swing和AWT库来实现。

## 业务逻辑层的分解

业务逻辑层的开发包图请查看体系结构设计文档。

### logIn模块

1. 模块概述

logIn模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

logIn模块的职责和接口参见软件系统结构描述文档内对该模块的描述。

1. 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。在展示层和业务逻辑层之间添加logInBLService接口。为了隔离业务逻辑职责和逻辑控制职责，添加logInController，这样logInController会将对用户登录和客户注册的业务逻辑处理委托给logIn对象。guestPO、hotelWorkerPO、webMarketer、webManager是作为账户信息的持久性对象被添加到设计模型中去的。

1. 模块内部类的接口规范
2. Controller类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| logInController.guestLogIn | 语法 | Public ResultMessage guestLogIn (String guest, String password) |
| 前置条件 | 客户输入客户编号和密码 |
| 后置条件 | 客户被认证并进入系统客户主界面 |
| logInController.hotelWorkerLogIn | 语法 | Public ResultMessage hotelWorkerLogIn (String hotelWorker, String password) |
| 前置条件 | 酒店工作人员输入编号和密码 |
| 后置条件 | 酒店工作人员被认证并进入系统酒店工作人员主界面 |
| logInController.webMarketerLogIn | 语法 | Public ResultMessage webMarketerLogIn (String webMarketer, String password) |
| 前置条件 | 网站营销人员输入编号和密码 |
| 后置条件 | 网站营销人员被认证并进入系统网站营销人员主界面 |
| logInController.webManagerLogIn | 语法 | Public ResultMessage webManagerLogIn (String webManager, String password) |
| 前置条件 | 网站管理人员输入编号和密码 |
| 后置条件 | 网站管理人员被认证并进入系统网站管理人员主界面 |
| logInController.guestSignUp | 语法 | Public ResultMessage guestSignUp (GuestVO guestVO) |
| 前置条件 | 客户信息符合规范要求 |
| 后置条件 | 系统返回是否成功添加客户；是则进入系统的客户主界面，否则提示添加失败 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| logIn.guestLogIn | 客户登录系统 | |
| logIn.hotelWorkerLogIn | 酒店工作人员登录系统 | |
| logIn.webMarketerLogIn | 网站营销人员登录系统 | |
| logIn.webManagerLogIn | 网站管理人员登录系统 | |
| logIn.guestSignUp | 客户注册 | |

1. Domain类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| logIn.guestLogIn | 语法 | Public ResultMessage guestLogIn (String guest, String password) |
| 前置条件 | 客户输入客户编号和密码 |
| 后置条件 | 客户被认证并进入系统客户主界面 |
| logIn.hotelWorkerLogIn | 语法 | Public ResultMessage hotelWorkerLogIn (String hotelWorker, String password) |
| 前置条件 | 酒店工作人员输入编号和密码 |
| 后置条件 | 酒店工作人员被认证并进入系统酒店工作人员主界面 |
| logIn.webMarketerLogIn | 语法 | Public ResultMessage webMarketerLogIn (String webMarketer, String password) |
| 前置条件 | 网站营销人员输入编号和密码 |
| 后置条件 | 网站营销人员被认证并进入系统网站营销人员主界面 |
| logIn.webManagerLogIn | 语法 | Public ResultMessage webManagerLogIn (String webManager, String password) |
| 前置条件 | 网站管理人员输入编号和密码 |
| 后置条件 | 网站管理人员被认证并进入系统网站管理人员主界面 |
| logIn.guestSignUp | 语法 | Public ResultMessage guestSignUp (GuestVO guestVO) |
| 前置条件 | 客户信息符合规范要求 |
| 后置条件 | 系统返回是否成功添加客户；是则进入系统的客户主界面，否则提示添加失败 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| User.getSingle(String userID,UserType userType) | 返回与该ID对应的user信息 | |
| User.add(UserVo newUserVo) | 新增该用户并持久化保存该用户的数据 | |
| User.getLogInfo(String userID, UserType userType) | 返回与该ID对应的user密码 | |

1. 业务逻辑层的动态模型
2. 业务逻辑层的设计原理

## 数据层的分解

数据层主要给业务逻辑层提供数据访问服务，包括对于持久化数据的增、删、改、查。例如，AccountBusinessService的服务主要由AccountDataService提供。由于持久化数据的保存可能存在多种形式：Txt文件、序列化文件、数据库等，所示抽象了数据服务。以AccountDataService为例，数据层模块的具体描述下图所示：

### 数据层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| orderData | 负责保存订单信息，进行增、改、查的操作 |
| hotelData | 负责保存酒店信息，进行增、改、查的操作 |
| promotionData | 负责保存促销策略信息，进行增、删、改、查的操作 |
| marketData | 负责保存会员等级制度信息，进行改、查的操作 |
| guestData | 负责保存客户信息，进行增、改、查的操作 |
| hotelWorkerData | 负责保存酒店工作人员信息，进行增、改、查的操作 |
| webMarketerData | 负责保存网站营销人员信息，进行改、查的操作 |
| webManagerData | 负责保存网站管理人员信息，进行改、查的操作 |

### 数据层模块的接口规范

1. **orderData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| orderData.createOrder | 语法 | Public ResultMessage createOrder (OrderPO order) |
| 前置条件 | 不存在相同的orderId |
| 后置条件 | 数据库中添加一条订单数据 |
| orderData.getOrderDetails | 语法 | Public OrderPO getOrderDetail (String orderID) |
| 前置条件 | 无 |
| 后置条件 | 系统返回对应id的OrderPO,如果id不存在，返回Null |
| orderData.getAllGuestOrderGeneral | 语法 | Public List<OrderGeneralPO> getAllGuestOrderGeneral (GuestID GuestID) |
| 前置条件 | 无 |
| 后置条件 | 系统返回对应id的OrderPO,如果id不存在，返回Null |
| orderData. getAllHotelOrderGeneral | 语法 | Public List<OrderGeneralPO> getAllHotelOrderGeneral (GuestID GuestID) |
| 前置条件 | 无 |
| 后置条件 | 系统返回对应id的OrderPO,如果id不存在，返回Null |
| orderData. executeOrder | 语法 | Public ResultMessage executeOrder (String orderID) |
| 前置条件 | 存在相同的orderId且订单状态为未执行 |
| 后置条件 | 数据库中更改未执行订单的状态为已执行 |
| orderData. undoAbnormalOrder | 语法 | Public ResultMessage undoAbnormalOrder (String orderID) |
| 前置条件 | 存在相同的orderId且订单状态为异常 |
| 后置条件 | 数据库中更改并保存订单的状态数据 |
| orderData. undoNormalOrder | 语法 | Public ResultMessage undoNormalOrder (String orderID) |
| 前置条件 | 存在相同的orderId且订单状态为未执行 |
| 后置条件 | 数据库中更改并保存订单的状态 |

1. **hotelData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| hotelData.getHotelList | 语法 | Public List<HotelGeneralPO> getHotelList(AddressPO addressPO) |
| 前置条件 | 已知商圈和地址信息 |
| 后置条件 | 系统在数据库中查询并返回符合该商圈和地址信息的所有酒店列表 |
| hotelData.getHotelDetail | 语法 | Public HotelPO getHotelDetail(String hotelID) |
| 前置条件 | 已知客户点击查看的酒店编号 |
| 后置条件 | 返回保存酒店详细信息的HotelPO对象 |
| hotelData.addEvaluation | 语法 | Public boolean addEvaluation (EvaluationPO evaluationPO) |
| 前置条件 | 已知评论信息的对象evaluationPO以及被评论的酒店编号 |
| 后置条件 | 系统向数据库中添加evaluationPO，并返回添加是否成功的布尔值 |
| hotelData.getHotelInfo | 语法 | Public HotelInfoPO getHotelInfo (String hotelID) |
| 前置条件 | 已知酒店编号，逻辑层想获取酒店的基本信息 |
| 后置条件 | 返回一个保存有该酒店基本信息的HotelInfoPO对象 |
| hotelData.setHotelInfo | 语法 | Public boolean  setHotelInfo(HotelInfoPO hotelInfoPO) |
| 前置条件 | 已知保存有酒店基本信息的HotelInfoPO对象，逻辑层请求修改该酒店的基本信息 |
| 后置条件 | 系统在数据库中更新该HotelInfoPO对象，并返回一个修改是否成功的布尔值 |
| hotelData.getHotelRoomInfo | 语法 | Public HotelRoomInfoPO getHotelRoomInfo(String hotelID) |
| 前置条件 | 已知酒店编号，逻辑层请求查看该酒店的客房信息 |
| 后置条件 | 系统返回一个保存有酒店客房信息的HotelRoomInfoPO的对象 |
| hotelData.setHotelRoomInfo | 语法 | Public boolean setHotelRoomInfo  (HotelRoomInfoPO hotelRoomInfoPO) |
| 前置条件 | 已知一个保存有酒店客房信息的HotelRoomInfoPO对象，逻辑层请求更新数据库的酒店客房信息 |
| 后置条件 | 系统更新数据库里的HotelRoomInfoPO对象，并返回是否修改成功的布尔值 |
| hotelData.setCheckInInfo | 语法 | Public boolean setCheckInInfo (CheckInInfo checkInInfo) |
| 前置条件 | 已知保存有入住的信息的对象，该订单已被执行，需要修改数据库中的入住信息 |
| 后置条件 | 系统修改数据库中的入住信息，并返回一个修改是否成功的布尔值 |
| hotelData.setCheckOutInfo | 语法 | Public boolean setCheckOutInfo (CheckOutInfo checkOutInfo) |
| 前置条件 | 已知保存有退房信息的对象，该客户已办理退房，需要更新数据库里的退房信息 |
| 后置条件 | 系统更新数据库里的退房信息，并返回更新是否成功的布尔值 |
| hotelData.getRemainRoomInfo | 语法 | Public RemainRoomInfoPO getRemainRoomInfo(String hotelID) |
| 前置条件 | 已知酒店的编号，逻辑层请求当前酒店剩余客房的信息 |
| 后置条件 | 系统返回一个保存剩余客房信息的RemainRoomInfoPO对象 |
| hotelData.setRemainRoomInfo | 语法 | Public boolean setRemainRoomInfo  (RemainRoomInfoPO remainRoomInfoPO) |
| 前置条件 | 已知保存有已被修改剩余客房信息的RemainRoomInfo的对象以及当前酒店的编号，逻辑层请求修改此酒店的剩余客房信息 |
| 后置条件 | 系统更新数据库里此酒店的剩余客房信息，并返回一个是否修改成功的布尔值 |
| hotelData.add(HotelInfoPO hotelInfoPO) | 语法 | Public boolean add (HotelInfoPO hotelInfoPO) |
| 前置条件 | 已知需要被添加酒店的HotelInfoPO对象 |
| 后置条件 | 系统向数据库里添加此HotelInfoPO对象，并返回一个添加是否成功的布尔值 |

1. **promotionData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| promotionData.getPromotions | 语法 | Public List<PromotionPO> getPromotions(String userID) |
| 前置条件 | 已知用户编号，逻辑层请求获得该用户（酒店工作人员、网站营销人员）所管理的促销策略 |
| 后置条件 | 返回该用户所管理的促销策略 |
| promotionData.setPromotions | 语法 | Public boolean setPromotions(String userID,List<PromotionPO> list) |
| 前置条件 | 已知用户编号和被修改的促销策略列表，逻辑层请求更新数据库中的策略列表 |
| 后置条件 | 系统更新数据库中的策略列表，并返回一个是否修改成功的布尔值 |

1. **marketData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| marketData.getMemberFormulation | 语法 | Public MarketPO getMemberFormulation() |
| 前置条件 | 无 |
| 后置条件 | 返回现有的会员等级制度信息 |
| marketData.setMemberFormulation | 语法 | Public MessageResult setMemberFormulation (HotelInfoPO hotelInfoPO) |
| 前置条件 | 网站营销人员输入会员等级制度信息 |
| 后置条件 | 系统记录此会员等级制度信息 |

1. **guestData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| guestData.modifyMember | 语法 | Public ResultMessage modifyMember(MemberPO memberPO) |
| 前置条件 | 无 |
| 后置条件 | 根据提供的memberPO修改单一持久化客户对象的信用值 |
| guestData.getSingle | 语法 | Public ResultMessage getSingle (String guestID) |
| 前置条件 | guestID.Length=客户的ID长度10 |
| 后置条件 | 系统返回与该ID对应的客户信息 |
| guestData.add | 语法 | Public ResultMessage add(GuestPO newGuestPO) |
| 前置条件 | 输入的guestID不存在，password和所有客户信息符合规范 |
| 后置条件 | 系统新增该客户并持久化保存该客户的数据 |
| guestData.modify | 语法 | Public ResultMessage modify  (GuestPO guestPO) |
| 前置条件 | 客户信息符合规则且齐全 |
| 后置条件 | 系统修改该用户信息并持久化保存该用户数据 |
| guestData.init | 语法 | Public ResultMessage init() |
| 前置条件 | 无 |
| 后置条件 | 清空所有客户持久化对象 |

1. **hotelWorker模块的接口规范**

|  |  |  |
| --- | --- | --- |
| hotelWorkerData.getSingle | 语法 | Public ResultMessage getSingle (String hotelWorkerID) |
| 前置条件 | hotelWorkerID.Length=酒店工作人员的ID长度8 |
| 后置条件 | 系统返回与该ID对应的酒店工作人员信息 |
| hotelWorkerData.add | 语法 | Public ResultMessage add(HotelWorkerPO newHotelWorkerPO) |
| 前置条件 | 输入的hotelWorkerID不存在，password和所有酒店工作人员信息符合规范 |
| 后置条件 | 系统新增该酒店工作人员并持久化保存该酒店工作人员的数据 |
| hotelWorkerData.modify | 语法 | Public ResultMessage modify  (HotelWorkerPO hotelWorkerPO) |
| 前置条件 | 酒店工作人员信息符合规则且齐全 |
| 后置条件 | 系统修改该酒店工作人员信息并持久化保存该酒店工作人员数据 |
| hotelWorkerData.init | 语法 | Public ResultMessage init() |
| 前置条件 | 无 |
| 后置条件 | 清空所有酒店工作人员持久化对象 |

1. **webMarketer模块的接口规范**

|  |  |  |
| --- | --- | --- |
| webMarketerData.getSingle | 语法 | Public ResultMessage getSingle (String webMarketID) |
| 前置条件 | webMarketID.Length=网站营销人员的ID长度6 |
| 后置条件 | 系统返回与该ID对应的网站营销人员信息 |
| webMarketerData.add | 语法 | Public ResultMessage add(WebMarketerPO newWebMarketerPO) |
| 前置条件 | 输入的webMarketerID不存在，password和所有网站营销人员信息符合规范 |
| 后置条件 | 系统新增该网站营销人员并持久化保存该网站营销人员的数据 |
| webMarketerData.modify | 语法 | Public ResultMessage modify  (WebMarketerPO webMarketerPO) |
| 前置条件 | 网站营销人员信息符合规则且齐全 |
| 后置条件 | 系统修改该网站营销人员信息并持久化保存该网站营销人员数据 |
| webMarketerData.init | 语法 | Public ResultMessage init() |
| 前置条件 | 无 |
| 后置条件 | 清空所有网站营销人员持久化对象 |

1. **webManager模块的接口规范**

|  |  |  |
| --- | --- | --- |
| webManagerData.getSingle | 语法 | Public ResultMessage getSingle (String webManagerID) |
| 前置条件 | webManagerID.Length=网站管理人员的ID长度4 |
| 后置条件 | 系统返回与该ID对应的网站管理人员信息 |
| webManagerData.add | 语法 | Public ResultMessage add(WebManagerPO newWebManagerPO) |
| 前置条件 | 输入的webManagerID不存在，password和所有网站管理人员信息符合规范 |
| 后置条件 | 系统新增该网站管理人员并持久化保存该网站管理人员的数据 |
| webManagerData.modify | 语法 | Public ResultMessage modify  (WebManagerPO webManagerPO) |
| 前置条件 | 网站管理人员信息符合规则且齐全 |
| 后置条件 | 系统修改该网站管理人员信息并持久化保存该网站管理人员数据 |

# 信息视角

## 数据持久化对象

系统的PO类就是对应的相关的实体类，如下所示。

|  |  |
| --- | --- |
| 类名 | 包含的属性 |
| MemberPO | 客户编号、生日（普通会员）、企业名称（企业会员） |
| GuestPO | 客户编号、生日（普通会员）、企业名称（企业会员）、姓名、昵称、密码、联系方式，信用值 |
| CreditPO | 客户编号、信用记录（时间、订单号、动作、信用度变化、信用度结果） |
| HotelWorkerPO | 酒店／酒店工作人员编号、密码、所属酒店 |
| WebMarketPO | 网站营销人员编号、密码 |
| WebManagerPO | 网站管理人员编号、密码 |
|  |  |
| OrderGeneralPO | 订单编号、客户编号、酒店编号、酒店名、地址、价格、入住日期和退房日期、订单状态 |
| OrderPO | 订单编号、客户编号、酒店编号、酒店名、地址、价格、生成时间、入住时间和退房时间、最晚订单执行时间（注：客户填写）、订单状态、房间类型、房间数、预计入住人数、入住人姓名、联系方式、特别要求（如有无儿童）、房间号、预计离开时间（注：酒店工作人员填写）、实际离开时间（注：酒店工作人员填写） |
| CheckInPO | 订单编号、房间号，入住时间、预计离开时间 |
| CheckOutPO | 订单编号、实际离开时间 |
|  |  |
| HotelGeneralPO | 酒店编号、酒店名称、所属城市商圈、星级、评分、最低房价 |
| HotelPO | 酒店编号、酒店编号、酒店名称、所属城市商圈、详细地址、星级、评分评价、简介、设施服务 |
| RoomInfoPO | 酒店编号、类型、数量、原始价格 |
| RemainRoomInfoPO | 酒店编号、类型、数量、原始价格、日期 |
|  |  |
| MarketPO | 等级名称、等级信用度、等级优惠 |
|  |  |
| PromotionPO | 促销策略、折扣率、折扣酒店、开始日期和结束日期 |

## 文件格式

本系统依据不同情况分别采用数据库、序列化、txt文件持久化保存三种方式保存文件。

|  |  |  |
| --- | --- | --- |
| 所属包与表名 | 内容 | 字段设计 |
| Guest.creditRecord | 信用记录 |  |
| Guest.creditValues | 信用值 |  |
| Guest.guestInfo | 客户信息 |  |
| Guest.memberInfo | 会员信息 |  |
| HotelWorker.  hotelWorkerInfo | 酒店工作人员信息 |  |
| Webmarketer.  webMarketerInfo | 网站营销人员信息 |  |
| WebManager.  webManagerInfo | 网站管理人员信息 |  |
| Order.hotelInfo | 酒店相关信息 |  |
| Oder.orderInfo | 订单信息 |  |
| Order.orderGeneral | 订单概况 |  |
| Order.roomInfo | 房间信息 |  |
| Order.timeInfo | 订单相关时间信息 |  |
| Order.userInfo | 客户信息 |  |
| Promotion.  guestDiscountInfo | 客户折扣信息 |  |
| Promotion.  webPromotion | 网站促销策略 |  |
| Promotion.  hotelPromotion | 酒店促销策略 |  |
| Hotel.hotelInfo | 酒店信息 |  |
| Hotel.roomInfo | 客房信息 |  |
| Market.  memberRegulations | 会员等级制度 |  |