BLService可以传入VO

DataService只与PO打交道

需要一个统一的界面吗

**涉及列表的添加一个label显示记录总数**

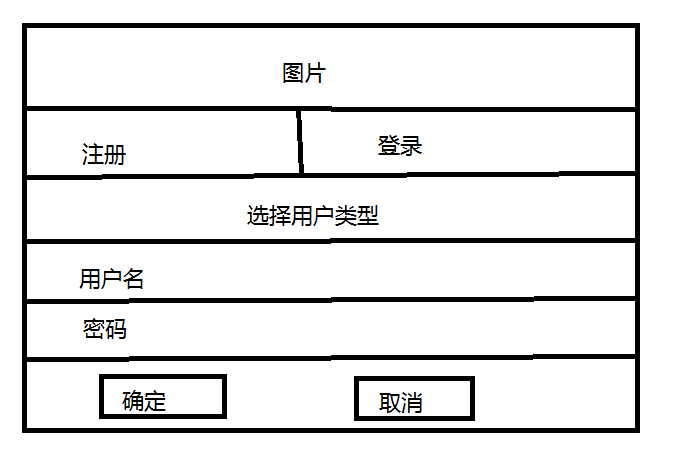
Hotel和OfflineRecord有一个共同的需求：找到给定日期、给定酒店的所有可用房间类型以及剩余房间数

RoomService依赖于AviableRoomService，二者共同返回这个结果。

模块1：User

包括用户注册和所有用户的登录

**LoginUI**



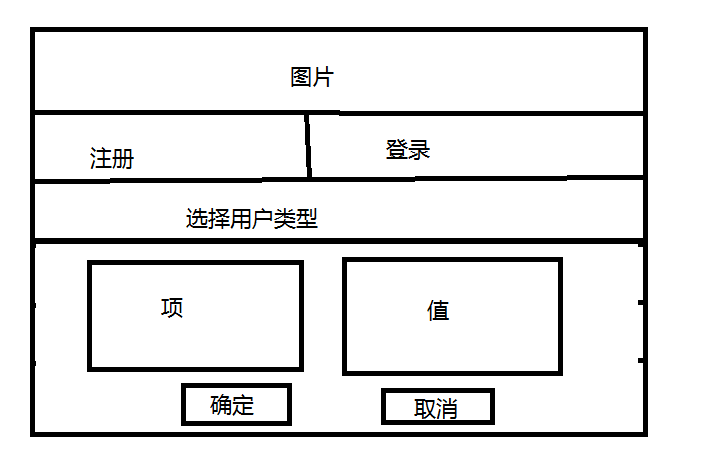
注册：选择提示：企业用户or普通用户

**企业用户ui**

**RegisterEnterpriseUI**

**普通用户ui**

**RegisterNormalUI**

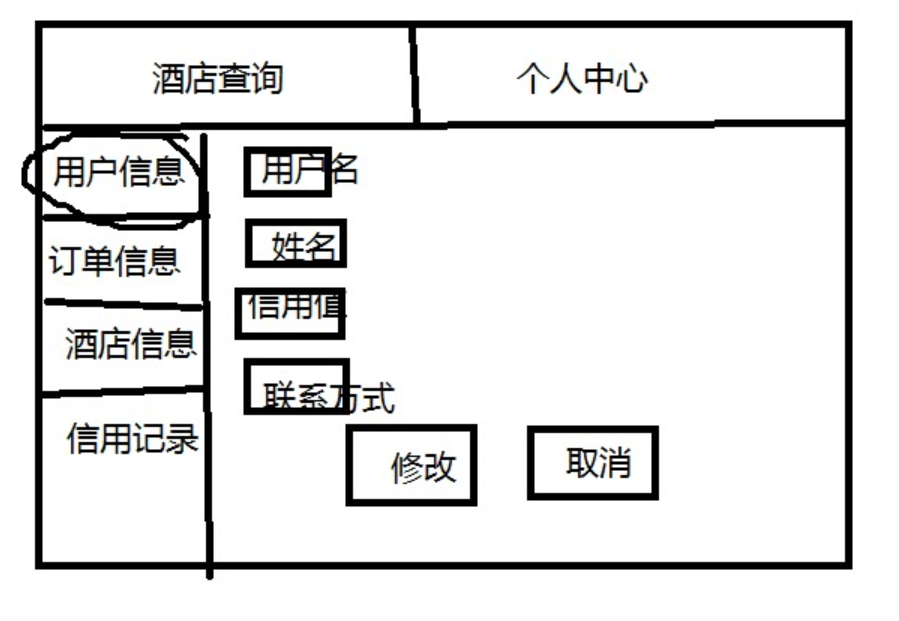


包括用户信息管理

网站管理人员管理用户信息

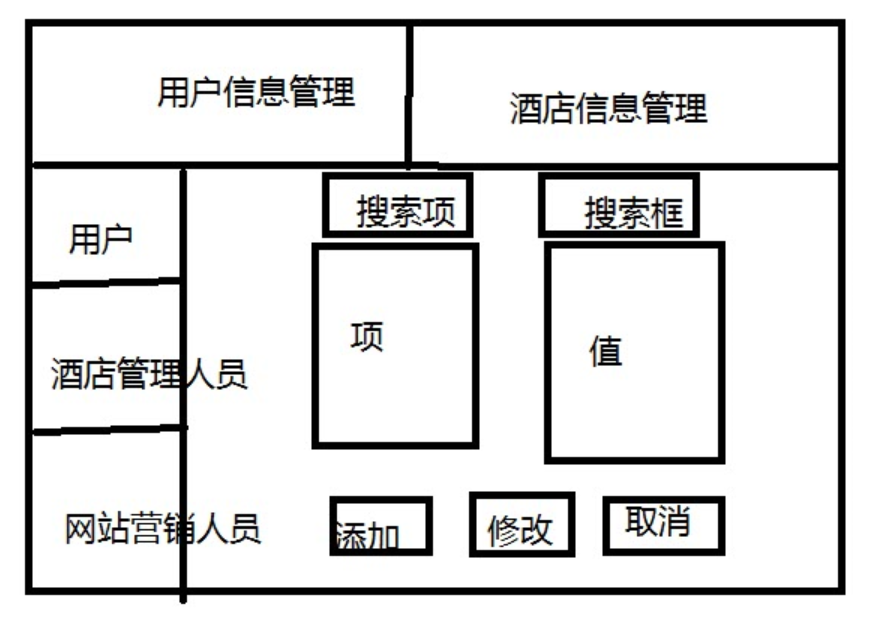
个人中心

**UserUI**



**网站管理人员的用户信息管理界面**

**WebUserUI**



**UserController持有UserService和StaffService**

public UserVO findByUsername(String username);

public ResultMessage register(UserVO)

public ResultMessage update(UserVO)

public UserVO userLogin(String username,String password);

public StaffVO staffLogin(String username,String password);

**------------------------------------------------**

**注意不能添加用户**

UserService

public UserVO findByUsername(String username);

public ResultMessage register(UserVO)

public ResultMessage update(UserVO)

**public UserVO login(String username,String password);**

**//注意登录时需要返回一个对象，因为要知道当前登录的对象的身份，只返回登录信息是不够的。如果非空，那么就算登录成功**

UserDAO

public UserPO findByUsername(String username)

public ResultMessage add(UserPO)

public ResultMessage update(UserPO)

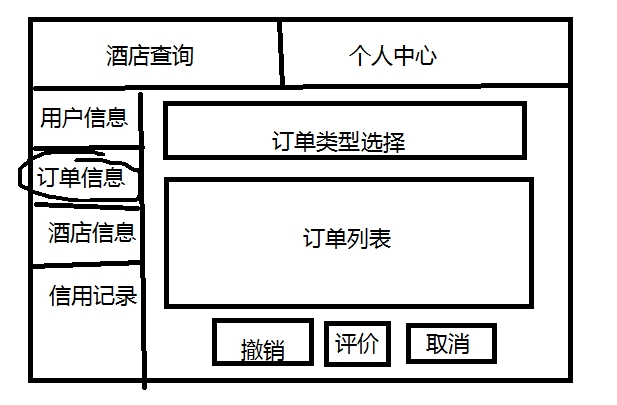
模块2：Order

包括用户订单（已预订酒店）的浏览，下单，撤销，评价

酒店的更新入住、退房、延迟入住信息，浏览订单

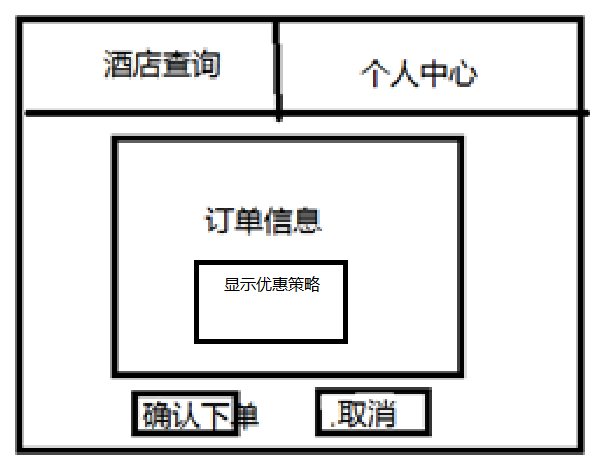
网站促销人员的浏览、撤销 异常订单

**UserOrderUI**

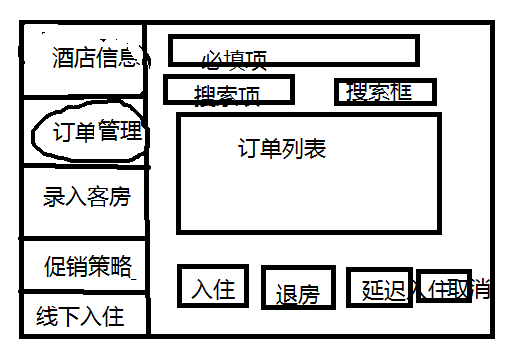


**PlaceOrderUI（可能是从HotelSearchUI和HotelDetailUI跳转到，注意跳转时调用两个display方法，其中一个方法返回的RoomVO中有可用房间数量）**

**（注意要展示优惠策略和减的金额，优惠策略不可叠加）**

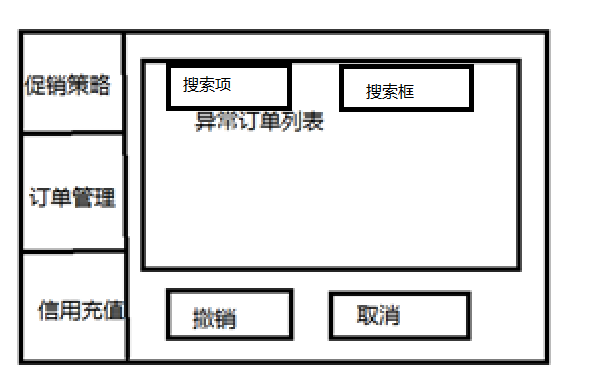


**HotelOrderUI**



**//按用户名或订单号得到用户信息和订单列表信息/订单信息**

**WebOrderUI**



**//增加搜索项和搜索框（按用户名和按订单号）**

**OrderController**

**持有一个OrderService和UserService（返回用户信息）**

OrderService

public

public OrderVO findByID(int ID);

public List<OrderVO> findByUsernameAndType(String username,OrderType)

**public List<OrderVO> findByUsername(String username)**

**public List<OrderVO> findByHotelAndUsername(int hotelID,String username);**

public List<OrderVO> findByOrderType(OrderType)

public List<OrderVO> findByHotelAndTime(int hotelID,Date begin,Date end);

public OrderVO placeOrder(OrderVO);

**//OrderVO添加2个属性，表示经过的降价处理。**

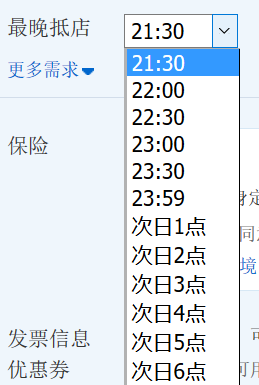


**//OrderVO信息不全，在方法内进行以下处理**

**//下单要依赖于读取优惠策略 HotelDiscountService（处理降价的逻辑，详细设计阶段处理）**

**//WebDiscountService**

**//下单要依赖于读取用户生日、所在企业、原始信用值、会员等级（UserService的findByUsername）**

**下单时要收集最晚订单执行时间（即执行时间），预计离开时间为离开当天的12点**

**public ResultMessage add(OrderVO);**

**public ResultMessage update(OrderVO)（包括用户撤销，用户评价，网站营销人员撤销；酒店管理人员更新入住、更新退房、更新延迟入住信息）**

**更新的同时要改变用户信用值，依赖于CreditRecordService的add()**

**注意new出CreditRecordVO时不需要设置原始信用值，由它自己来实现；网站营销人员撤销异常订单时也会改变用户信用值；入住时会更新信用值，延迟入住也会改变信用值**

OrderDAO

public OrderPO findByID(int ID);

public List<OrderPO> findByUsernameAndType(String username,Type)

**public List<OrderPO> findByHotelAndUsername(int hotelID,String username);**

public List<OrderPO> findByUsername(String username)

public List<OrderPO> findByOrderType(OrderType)

public List<OrderPO> findByHotelAndTime(int hotelID,Date begin,Date end);

public ResultMessage add(OrderPO);

public ResultMessage update(OrderPO)（包括用户撤销，用户评价，网站营销人员撤销）

模块3：Hotel

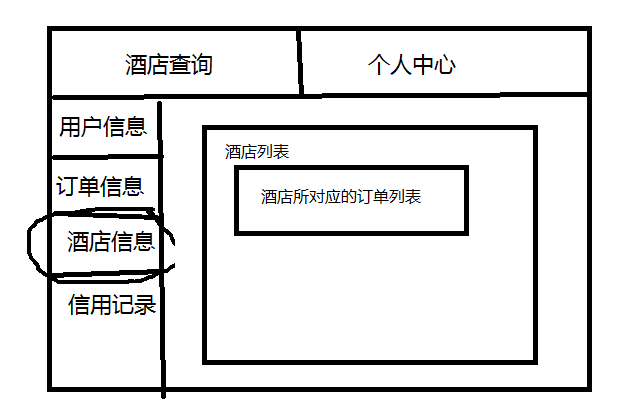
包括用户浏览已预订的酒店信息

搜索浏览酒店

酒店基本信息管理

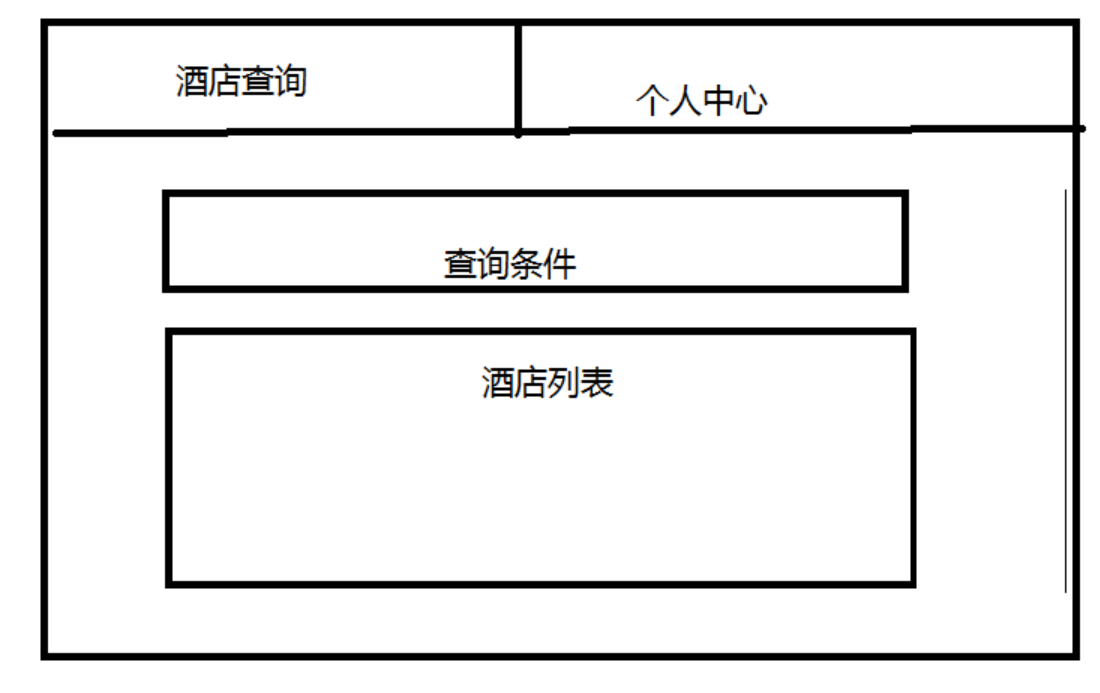
网站管理人员添加酒店

**UserHotelUI**



**//改 酒店及其对应订单**

**HotelSearchUI**

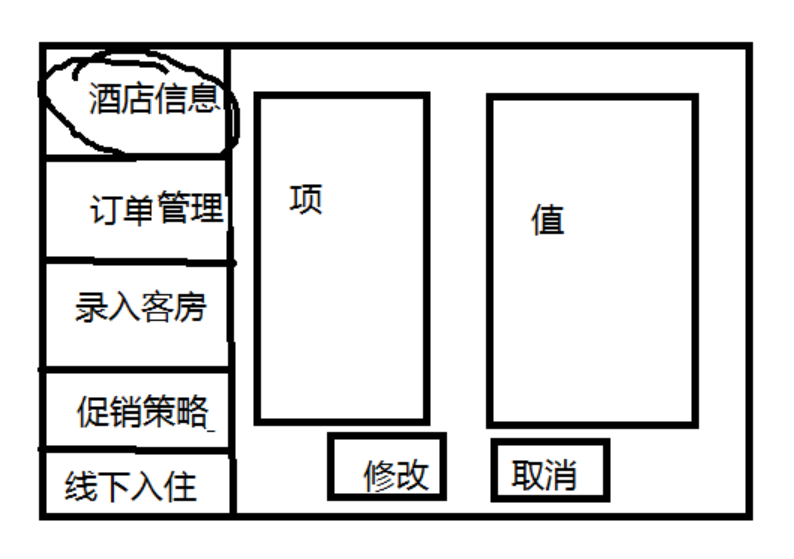


**HotelDetailUI**

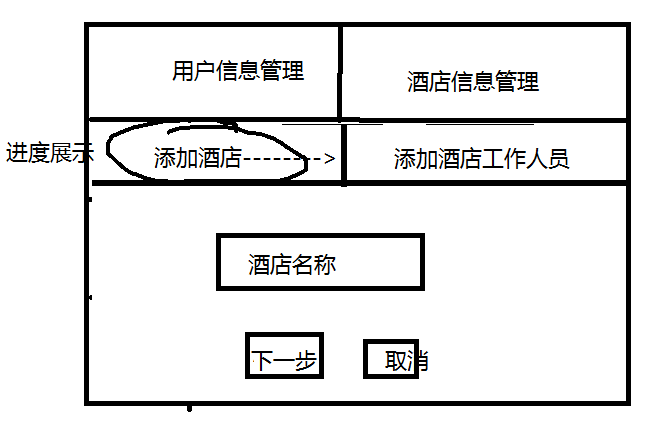


**//还要显示可以预定的房间信息（用例描述也要改）**

**HotelUI**



**WebHotelUI**



**//确定改成下一步**

HotelService:

public HotelVO findByID(int hotelID);

public ResultMessage update(HotelVO);

**public ResultMessage add(HotelVO);//只有在网站管理人员StaffService那里会添加酒店**

**public Map<HotelVO,List<OrderVO>> findOrderedHotelAndOrder(String**

**username);//返回用户曾预订的酒店列表以及每个酒店对应的订单信息，依赖于OrderService，并调用findByID**

**（要显示在UserHotel）**

**下单时会得到最晚执行时间和预计离开时间；用户撤销会去掉这条记录；线下的入住会得到执行时间和预计离开时间；订单状态置为异常也相当于撤销。**

public **Map<HotelVO,List<RoomVO>>** find (int loc,int circle,Date begin,Date end);

//查询结果不能直接显示，还需查订单情况

**//通过RoomService的findAvailableByHotelID(int hotelID,Date begin,Date end)，得到包含所有房间类型的可用房间数量（可订的房间，RoomService依赖于AvailableRoomService）**

**//通过OrderService的findByHotelAndUsername，给HotelVO中添加其List<OrderStatus>属性值**

//在上个查询基础上，在内存中过滤

public **Map<HotelVO,List<RoomVO>>**

filterName(**Map<HotelVO,List<RoomVO>>**,String name)

public **Map<HotelVO,List<RoomVO>>**

filterStar(**Map<HotelVO,List<RoomVO>>**,int star)

public **Map<HotelVO,List<RoomVO>>**

filterScore(**Map<HotelVO,List<RoomVO>>**,double low,double high);

public **Map<HotelVO,List<RoomVO>>**

orderByStar(**Map<HotelVO,List<RoomVO>>**,boolean isDecrease);

public **Map <HotelVO,List<RoomVO>>**

orderByScore(**Map<HotelVO,List<RoomVO>>**,boolean isDecrease);

public **Map.Entry<HotelVO,List<RoomVO>>**

**displayRoomDetail(Map<HotelVO,List<RoomVO>>,int hotelID)（当点击详细信息）**

（不丢弃这个数据结构对象，而是在此基础上进行过滤、排序）

**public Map.Entry<HotelVO,List<OrderVO>>**

**displayOrderDetail(HotelVO vo,String username)（当点击详细信息）**

**//依赖于OrderService的findByHotelAndUsername**

public **Map<HotelVO,List<RoomVO>>**

**filterIfOrdered(Map<HotelVO,List<RoomVO>>);**

**//直接根据HotelVO的属性来过滤**

public **Map<HotelVO,List<RoomVO>>**

**filterRoomType(Map<HotelVO,List<RoomVO>>,RoomType);**

public **Map<HotelVO,List<RoomVO>>**

**filterValue(Map<HotelVO,List<RoomVO>>,double low,double high);**

public **Map<HotelVO,List<RoomVO>>**

**orderByValue(Map<HotelVO,List<RoomVO>>);**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**不依赖于LocationService（Hotel查询时要用到）（同时也要显示在HotelUI上）**

**仅用于显示**

public List<LocationVO> findAll();

LocationDAO

public List<LocationPO> findAll();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**不依赖于CommCircleService（Hotel查询时要用到）（同时也要显示在HotelUI上）**

**仅用于显示**

public List<CommCircleVO> findByLoc(int locID);

CommCircleDAO

public List<CommCirclePO> findByLoc(int locID);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HotelDAO

public HotelPO findByID(int hotelID);

public ResultMessage update(HotelPO);

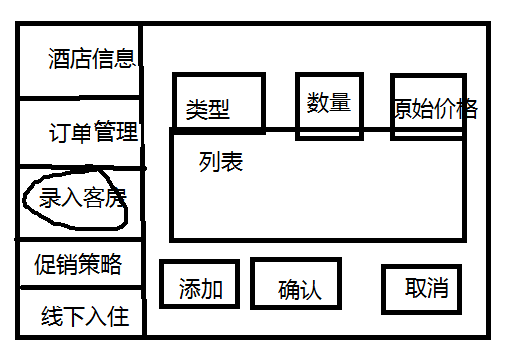
public ResultMessage add(HotelPO);

public **List<HotelPO >** find (int loc,int circle);

模块4：Room

录入可用客房

**RoomUI**



**//改为列表显示**

RoomService

public RoomVO findByHotelAndType(int hotelID,RoomType);

**public List<RoomVO> findAvailableByHotelID(int hotelID,Date begin,Date end);**

**//依赖于AvailableRoomService，给RoomVO设置availableRoomNum属性**

**List<RoomType> findNotAddedRoomType(int hotelID);**

**（去掉RoomType表，改为枚举，综合枚举和Room表，显示没有的类型）**

public ResultMessage add(RoomVO);

public ResultMessage update(RoomVO);

RoomDAO

public RoomPO findByHotelAndType(int hotelID,RoomType);

public ResultMessage update(RoomPO);

public ResultMessage add(RoomPO);

public List<RoomPO> findByHotelID(int hotelID);

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**AvailableRoomService**

**public int findAvailableRoomNum(int hotelID,RoomType type,Date begin,Date end);**

**AvailableRoomDAO**

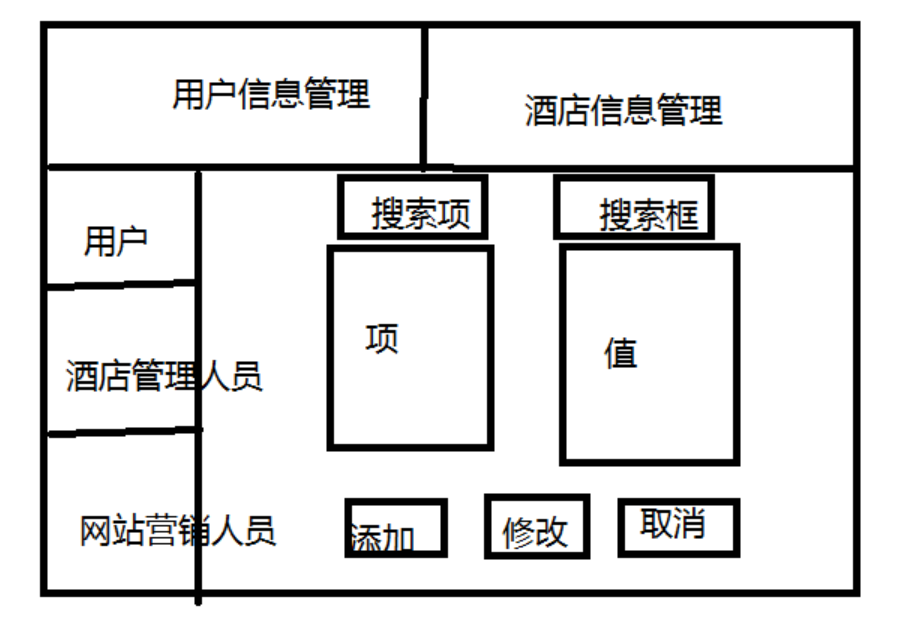
**public int findAvailableRoom(int hotelID,RoomType,Date date);**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

模块5：Staff

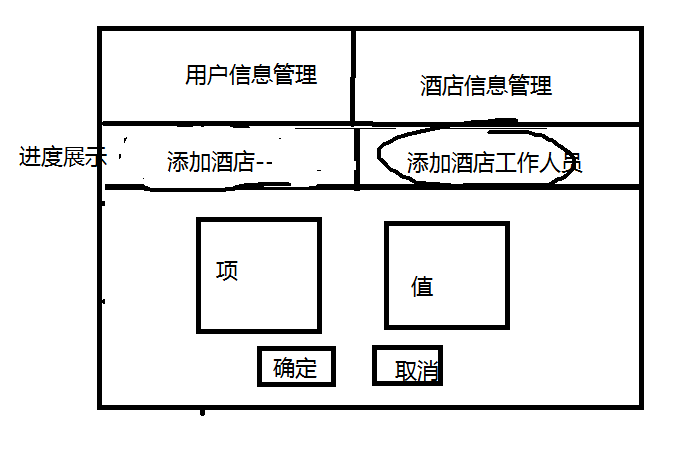
包括网站管理人员管理酒店工作人员、网站营销人员信息，添加酒店信息

**StaffUI（取消StaffUI，分为两个WebHotelStaffUI,WebMarketerStaffUI）**



**不能添加酒店管理人员信息，只能通过添加酒店实现（或者只有当选择网站营销人员时才显示添加按钮）**

**WebHotelAddUI**



**只有添加完酒店工作人员才更新酒店数据库和酒店工作人员数据库**

**调用酒店的add方法之后，取出酒店的id set给酒店工作人员**

**由Controller持有这两个Service的引用**

**StaffController 持有StaffService和HotelService**

StaffService

**public StaffVO login(String username,String password);//供User调用**

public ResultMessage update(StaffVO);

public ResultMessage add(StaffVO);

StaffDAO

public ResultMessage update(StaffPO);

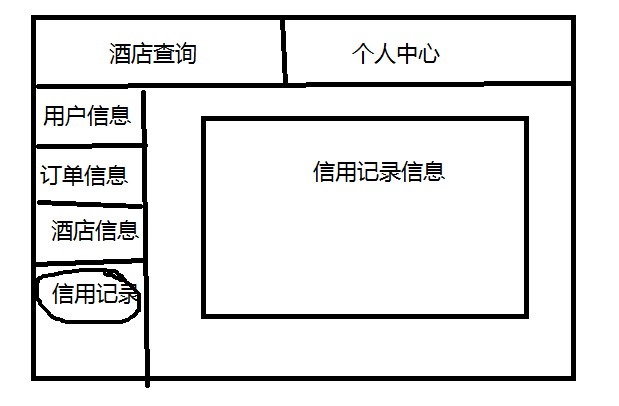
public ResultMessage add(StaffPO);

public StaffPO findByUsername(String username);

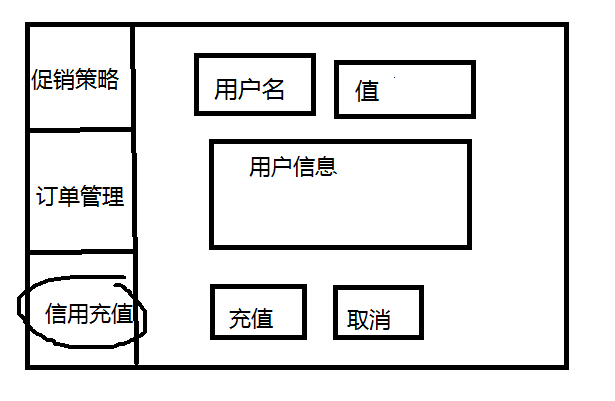
模块6：CreditRecord

包括用户查看信用记录，网站营销人员信用充值

**CreditRecordUI**



**WebCreditUI**



**//没有搜索项**

**CreditRecordController持有CreditRecordService和UserService**

CreditRecordService

public List<CreditRecordVO> findByUsername(String username);

**//由Controller持有UserService，负责展示用户信息**

public ResultMessage add(CreditRecordVO);**//传入的这个VO中的currCredit没有进行设置**

**//只供OrderService调用**

**//依赖于UserService的findByUsername，先获得原始信用值**

**//依赖于UserService的update，在添加信用记录的同时更新用户信用值**

**//注意取消创建用户时的Register这一状态**

CreditRecordDAO

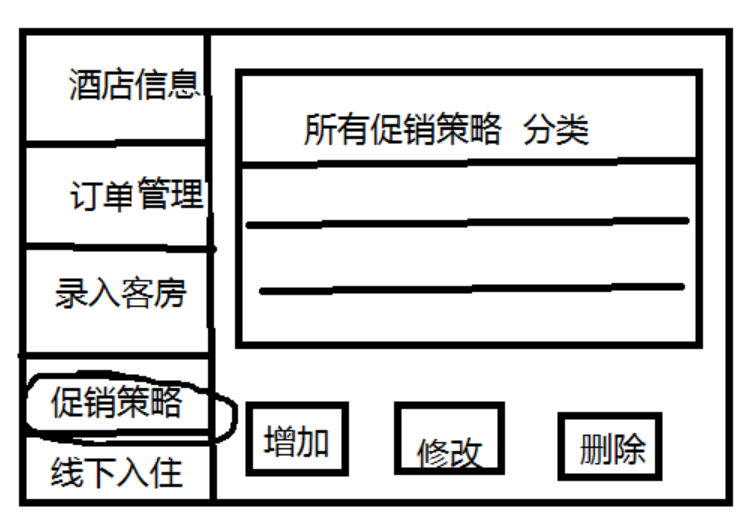
public List<CreditRecordPO> findByUsername(String username);

public ResultMessage add(CreditRecordPO);

模块7：HotelDiscount

**酒店工作人员 制定酒店促销策略**

**HotelDiscountUI**



**//部分策略不能重复出现，在点击添加时要先搜索该类型是否已存在**

**HotelDiscountController持有HotelDiscountService和EnterpriseService**

HotelDiscountService

public ResultMessage add (HotelDiscountVO);

public ResultMessage update(HotelDiscountVO)

public ResultMessage delete(int id);

public List<HotelDiscountVO> findAllByHotelID(int hotelID);

**public List<HotelDiscountType> findNotAddedType(int hotelID);**

**//需要企业列表，不依赖于EnterpriseService**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**//不依赖于企业EnterpriseService（也要显示在HotelDiscountUI界面上）**

**仅用于显示，由Controller负责管理Enterprise**

EnterpriseService

public List<EnterpriseVO> getAllEnterprises();

EnterpriseDAO

public List<EnterprisePO> findAll();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HotelDiscountDAO

public ResultMessage add (HotelDiscountPO);

public ResultMessage update(HotelDiscountPO)

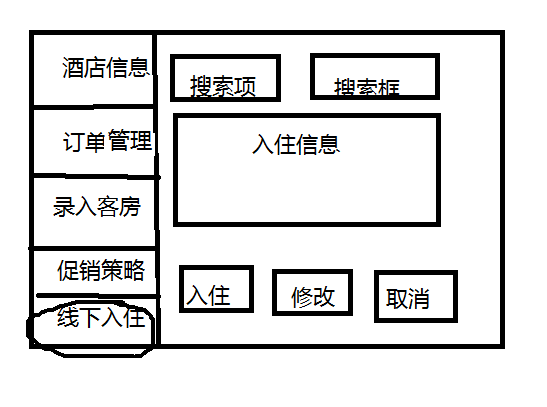
public ResultMessage delete(int id);

public List<HotelDiscountPO> findAllByHotelID(int hotelID);

模块8：OfflineRecord

酒店工作人员更新线下入住信息

**OfflineRecordUI**



**由Controller持有一个RoomService，得到当前酒店的所有可用房间（List<RoomVO>，其中也有可用房间数量）**

OfflineRecordService

public OfflineRecordVO findByID(int id);//线下退房时需要搜索

public ResultMessage add(OfflineRecordVO);

public ResultMessage update(OfflineRecordVO);

OfflineRecordDAO

public OfflineRecordPO findByID(int id);

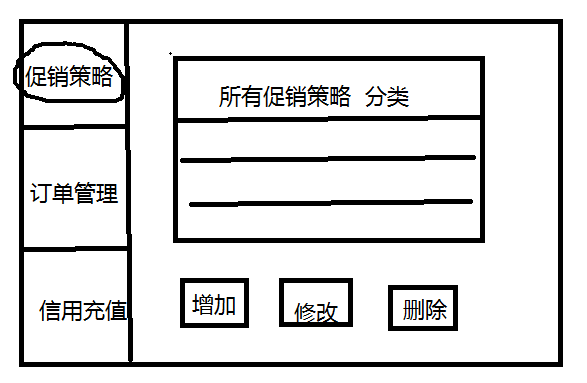
public ResultMessage add(OfflineRecordPO);

public ResultMessage update(OfflineRecordPO);

模板9：WebDiscount

**包括网站营销人员制定促销策略**

**WebDiscountUI**



WebDiscountService

public List<WebDiscountVO> findAll();

public ResultMessage add (WebDiscountVO);

public ResultMessage update(WebDiscountVO)

public ResultMessage delete(int id);

WebDiscountDAO

public ResultMessage add (WebDiscountPO);

public ResultMessage update(WebDiscountPO);

public ResultMessage delete(int id);

public List<WebDiscountPO> findAll();

按界面划分

**LoginUI（Login）**

**RegisterNormal（Login）**

**RegisterEnterprise（Login）**

-----------

用户

**UserUI（User）**

**UserOrderUI （Order）**

**UserHotelUI（Order）**

**PlaceOrderUI（Order）**

**HotelSearchUI（Hotel）**

**HotelDetailUI（Hotel）**

**CreditRecordUI（Credit）**

--------------

酒店管理人员

**HotelUI（Hotel）**

**RoomUI（Room）**

**HotelDiscountUI（HotelDiscount）**

**OfflineRecordUI（OfflineRecord）**

------------

网站营销人员

**WebOrderUI（Order）**

**WebCreditUI（Credit）**

**WebDiscountUI（WebDiscount）**

-----------

网站管理人员

**WebUserUI （User）**

**WebHotelUI（Hotel）**

**StaffUI（Staff）**

**HotelStaffUI（Staff）**