酒店预订系统

详细设计描述文档 v1.0

南京大学软件学院

45 组

刘宇翔 刘伟 刘宗侃 刘铮

2016.10.29

变更记录

修改人员	日期	变更原因	版本号
全体成员	2016.10.29	最初草稿	v1.0

1 / 84

目录

1. 引言

1.1. 编制目的

本报告详细完成对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而编写，是了解系统的导航。

1.2. 词汇表

词汇名称	词汇含义	备注
HRS	酒店预订系统	无

1.3. 参考资料

1) 酒店预订系统用例文档 v2.0

2) 酒店预订系统需求规格说明文档 v2.0

3) 酒店预订系统软件体系结构设计文档 v3.0

2. 产品描述

参考酒店预订系统用例文档和酒店预订系统需求规格说明文档中对产品的

概括描述。

3. 体系结构设计概述

参考酒店预订系统软件体系结构设计文档中对体系结构设计的概述。

4. 结构视角

4.1. 业务逻辑层的分解

业务逻辑层的开发包图参见软件体系结构设计文档图 4-1-1。

4.1.1. hotelbl 模块

4.1.1.1. 模块概述

hotelbl 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。

hotelbl 模块的职责及接口参见软件系统结构设计文档。

4.1.1.2. 整体结构

根据体系结构的设计，酒店预订系统选择分层体系结构风格，将系统分为 3 层（展示层、业务逻辑层、数据层）,很好地示意了整个高层抽象。展示层包含 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。每一层之间为了增加灵活性，添加了接口，比如在展示层和业务逻辑层之间添加 blservice.hotelblservice.HotelblService 接口，在业务逻辑层和数据层之间添加 dataservice.hoteldataservice.HotelDataService 接口。HotelPO 是作为酒店的持久化对象被添加到设计模型中去的。

hotelbl 模块的设计如图 4-1-1-1 所示。

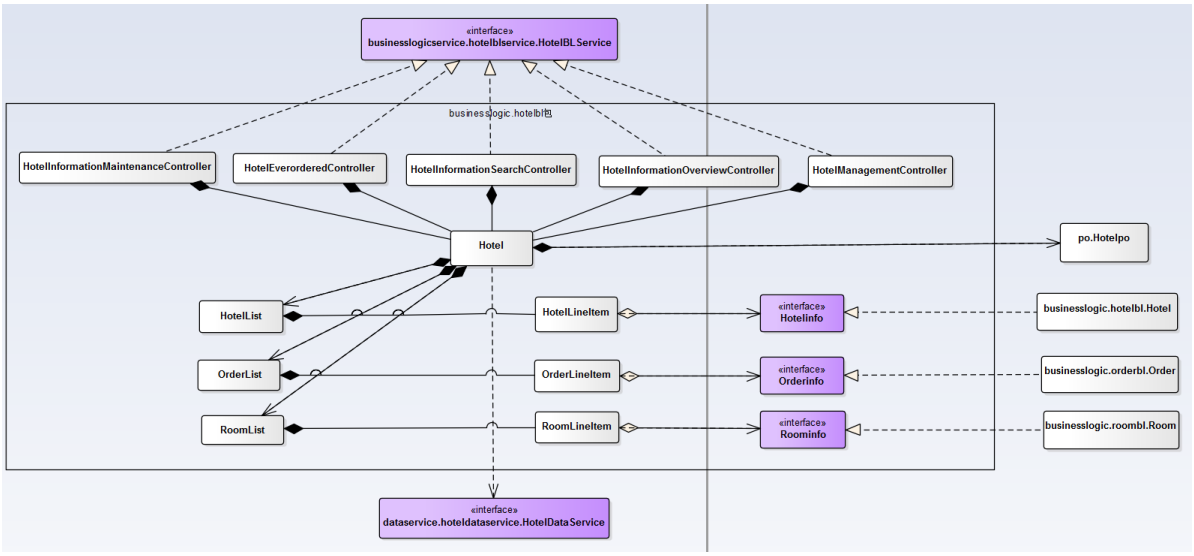

图 4-1-1-1　hotelbl 模块各个类的设计 hotelbl 模块各个类的职责如表 4-1-1-1 所示。

表 4-1-1-1 hotelbl 模块各个类的职责

类	职责
Hotel	酒店的领域模型对象，拥有酒店数据的帐号、名称、地址、商圈、星级、评分、简介、设施服务、联系方式，可以帮助完成查找酒店、预定酒店、评价酒店的任务
HotelInformationMaintenanceController	负责维护酒店的基本信息和更新酒店数据所需要的服务
HotelEverorderedController	负责查找酒店和用户的历史订单的酒店并返回历史酒店列表的服务
HotelInformationSearchController	负责查找符合条件的酒店的服务
HotelInformationOverviewController	负责显示酒店的详细信息的服务
HotelManagementController	负责酒店帐号管理的服务
HotelList	获取所有酒店列表

OrderList	获取所有订单列表
RoomList	获取所有房间列表
HotelLineItem	单个酒店的信息
OrderLineItem	单个订单的信息
RoomLineItem	单个房间的信息
Hotelinfo	负责获得酒店的对应信息的服务
Orderinfo	负责获得订单的对应信息的服务
Roominfo	负责获得房间的对应信息的服务

4.1.1.3. 模块内部类的接口规范

Hotelbl 的接口规范如表 4-1-1-2 所示。

Hotel 的接口规范

提供的服务（供接口）		
Hotel.messagelook	语法	public ResultMessage messagelook(HotelVO hvo)
	前置条件	用户已登录成功
	后置条件	查找地址和商圈内的酒店，返回范围内的酒店信息
Hotel.messagesearch	语法	public ResultMessage messagesearch(MessageInput in)
	前置条件	用户已登录成功
	后置条件	查找符合条件的酒店并返回酒店列表
Hotel.historylook	语法	public ResultMessage historylook(String id)
	前置条件	用户已登录成功

	后置条件	查找用户的历史订单的酒店并返回历史酒店列表
Hotel.messagemaintain	语法	public ResultMessage messagemaintain(MessageInput in)
	前置条件	酒店工作人员已登陆
	后置条件	系统修改酒店的基本信息
Hotel.accountadmin	语法	public ResultMessage accuntadmin(MessageInput in)
	前置条件	网站管理人员已登录
	后置条件	系统增加酒店帐号和名称以及该酒店工作人员的帐号
Hotel.setscore	语法	public boolean setscore(int score)
	前置条件	用户对酒店作出评分
	后置条件	系统增加评分记录，更新评分的均值
Hotel.setcomment	语法	public boolean setcomment(String comment)

	前置条件	用户对酒店作出评价
	后置条件	系统增加酒店的评价记录
Hotel.getroominfo	语法	public boolean getroominfo(HotelVO vo)
	前置条件	用户要查看酒店详细信息
	后置条件	系统显示剩余房间
Hotel.pricesort	语法	public HotelVO pricesort(ArrayList<Hotel> ah)
	前置条件	客户要求酒店按价格高低排序
	后置条件	显示价格由低到高的酒店列表
Hotel.starsort	语法	public HotelVO starsort(ArrayList<Hotel> ah)

	前置条件	客户要求酒店按星级排序
	后置条件	显示按照星级排序的酒店列表
Hotel.scoresort	语法	public HotelVO scoresort(ArrayList<Hotel> ah)
	前置条件	客户要求酒店按平均评分排序
	后置条件	显示平均评分由高到低的酒店列表
Hotel.gethistoryorder	语法	public HotelVO gethistoryorder(Hotel h)
	前置条件	客户要求查看酒店细节
	后置条件	系统显示此客户在该酒店的历史订单
Hotel.gethistoryhotel	语法	public HotelVO gethistoryhotel()
	前置条件	客户要求查看预定过的酒店
	后置条件	系统显示此客户的历史预定酒店

需要的服务（需接口）	
服务名	服务
DatabaseFactroy.getHotelDatabase	得到 Hotel 数据库的服务的引用
DatabaseFactroy.getRoomDatabase	得到 Room 数据库的服务的引用
HotelDataService.find(String location)	在数据区寻找符合在区域中的酒店
HotelDataService.search(MessageInput in)	在数据区寻找符合条件的酒店
DatabaseFactroy.getOrderDatabase	得到 Order 数据库的服务的引用
HotelDataService.getHistory(HotelPO po)	在数据区寻找该酒店的历史订单
HotelDataService.insert(HotelPO po)	插入单一持久化对象
HotelDataService.delete(HotelPO po)	删除单一持久化对象

HotelDataService.update(MessageInput in)	更新单一持久化对象

Promotion 的接口规范

提供的服务（供接口）		
Promotion.madebyhotel	语法	public ResultMessage madebyhotel (PromotionVO vo)
	前 置条件	酒店工作人员已登录成功
	后 置条件	系统录入此酒店营销策略，并发布
Promotion.madebyweb	语法	public ResultMessage madebyweb (PromotionVO vo)
	前 置条件	网站营销人员已登录成功
	后 置条件	系统录入此网站营销策略，并发布

Promotion.memberlevelmade	语法	public ResultMessage memberlevelmade(PromotionVO vo)
	前置条件	网站营销人员已登录成功
	后置条件	系统记录会员等级标准
Promotion.cancel	语法	public boolean cancel(Promotion promotion)
	前置条件	客户已登录
	后置条件	在系统中取消此订单

需要的服务（需接口）	
服务名	服务
DatabaseFactroy.getPromotionDatabase	得到 Promotion 数据库的服务的引用
DatabaseFactroy.getHotelDatabase	得到 Hotel 数据库的服务的引用
DatabaseFactroy.getOrderDatabase	得到 Order 数据库的服务的引用
DatabaseFactroy.getUserDatabase	得到 User 数据库的服务的引用
PromotionDataService.insert(PromotionPO po)	插入单一持久化对象
PromotionDataService.delete (PromotionPO po)	删除单一持久化对象
PromotionDataService.memberlevelinsert(MessageInput in)	在数据区添加会员等级制度
PromotionDataService.memberlevelupdate(MessageInput in)	在数据区更新会员等级制度

表 4-1-1-2 Hotelbl 的接口规范

4.1.1.4. 业务逻辑层的动态模型

图 4-1-1-2 表明了酒店预订系统中，在增加酒店的过程中输入酒店信息之后，账户管理业务逻辑处理的相关对象之间的协作。

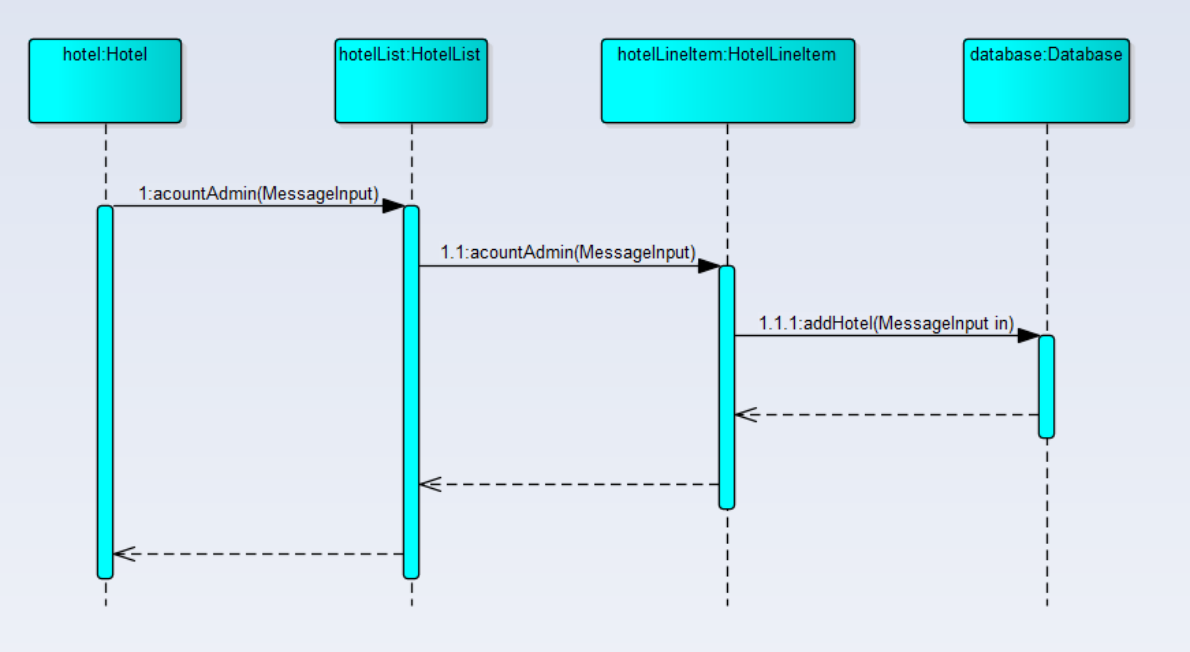

图 4-1-1-2 增加酒店的顺序图

图 4-1-1-3 表明了酒店预订系统中，在查看历史酒店的过程中输入一个 ID 之后，查看历史酒店业务逻辑处理的相关对象之间的协作。

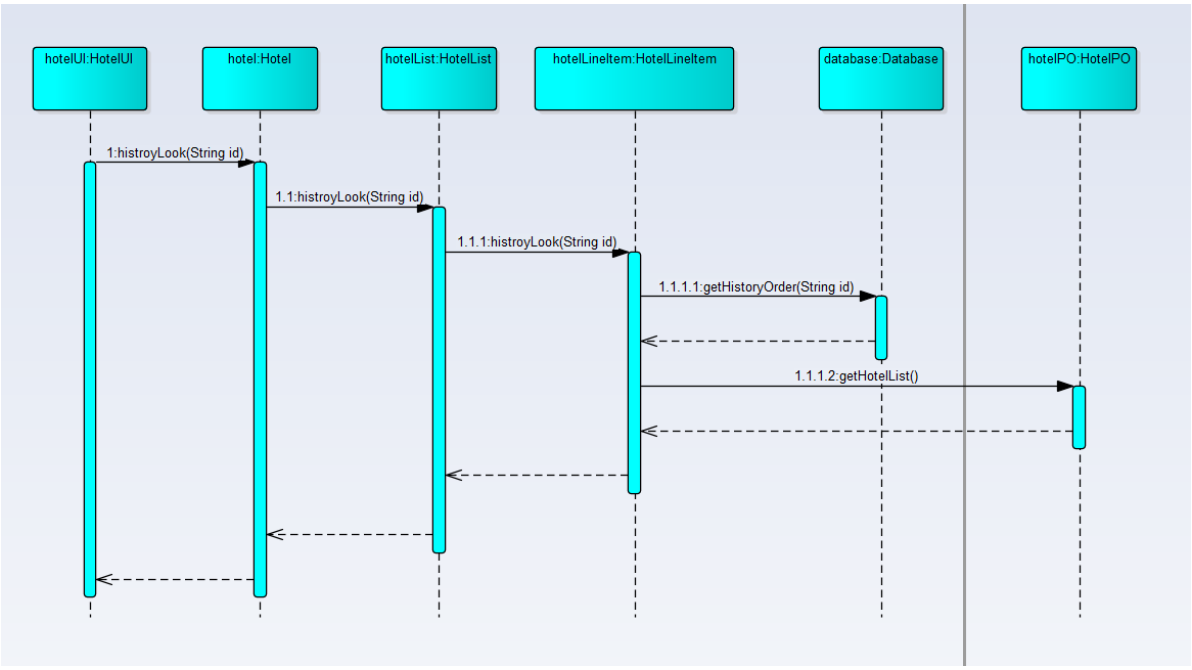

图 4-1-1-3 查看历史酒店的顺序图

图 4-1-1-4 表明了酒店预订系统中，在查看酒店细节的过程中输入一个酒店 VO

之后，查看历史订单业务逻辑处理的相关对象之间的协作。

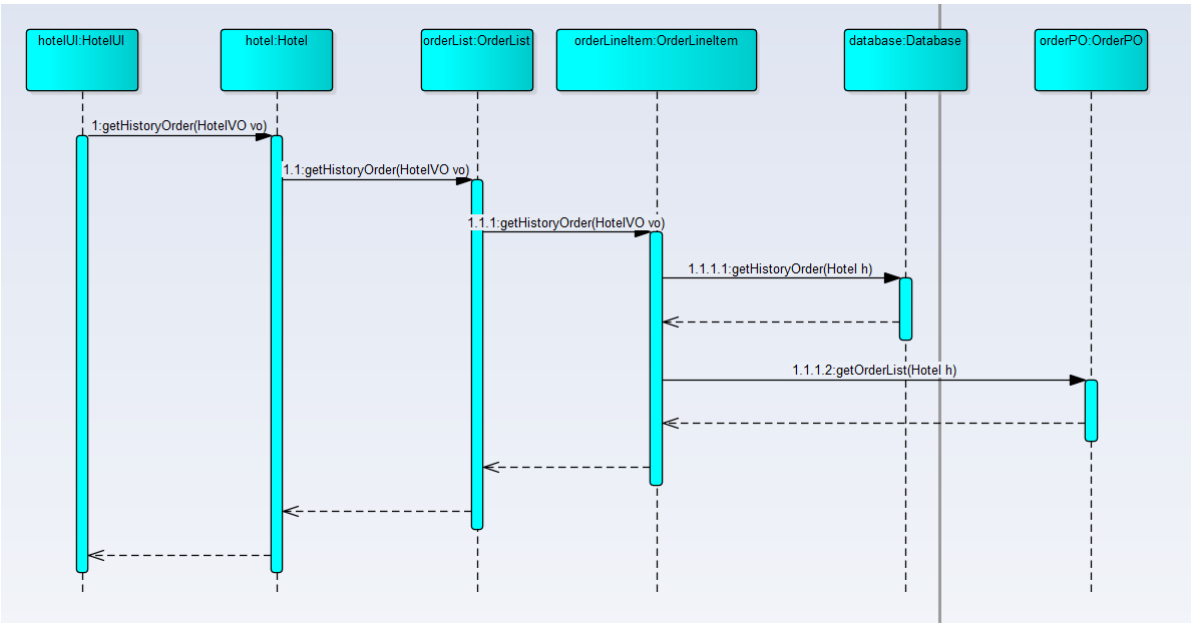

图 4-1-1-4 查看历史订单的顺序图

图 4-1-1-5 表明了酒店预订系统中，在查看房间信息的过程中输入一个酒店 VO

之后，查看房间信息业务逻辑处理的相关对象之间的协作。

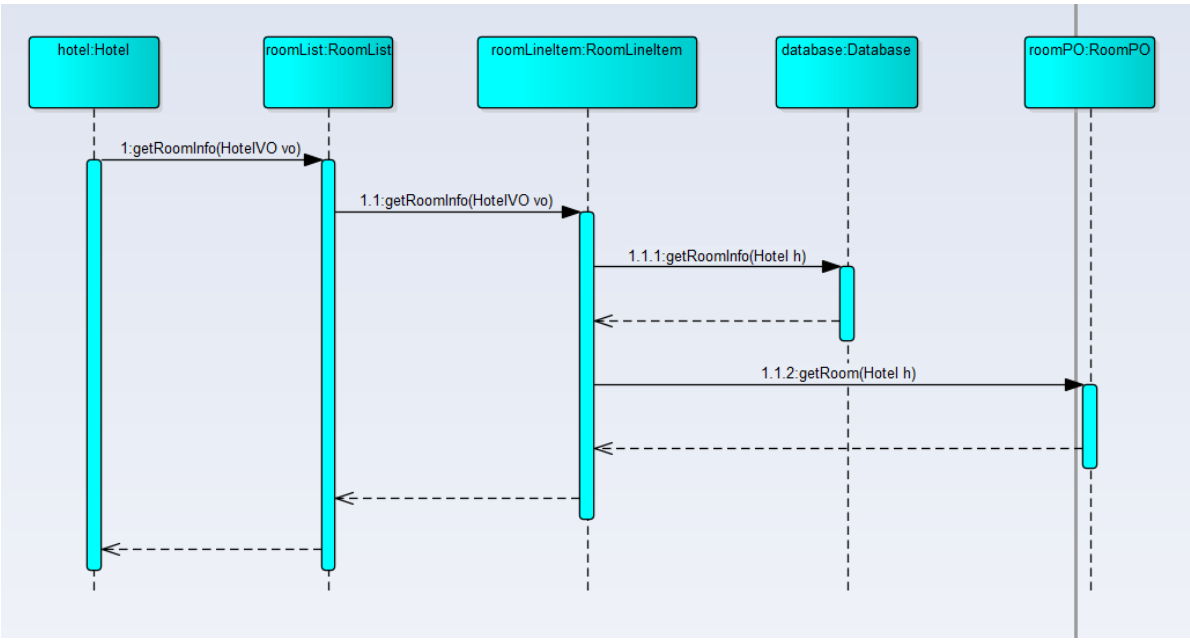

图 4-1-1-5 查看房间信息的顺序图

图 4-1-1-6 表明了酒店预订系统中，在浏览酒店的过程中输入一个酒店 VO 之后，浏览酒店业务逻辑处理的相关对象之间的协作。

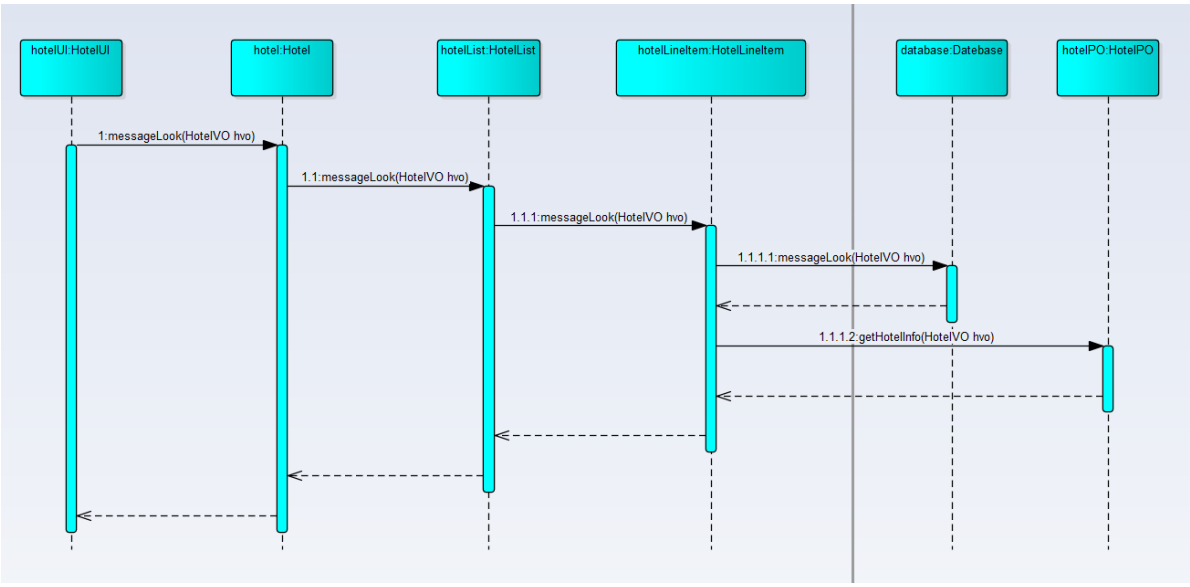

图 4-1-1-6 浏览酒店的顺序图

图 4-1-1-7 表明了酒店预订系统中，在维护酒店基本信息的过程中输入修改信息

之后，维护酒店业务逻辑处理的相关对象之间的协作。

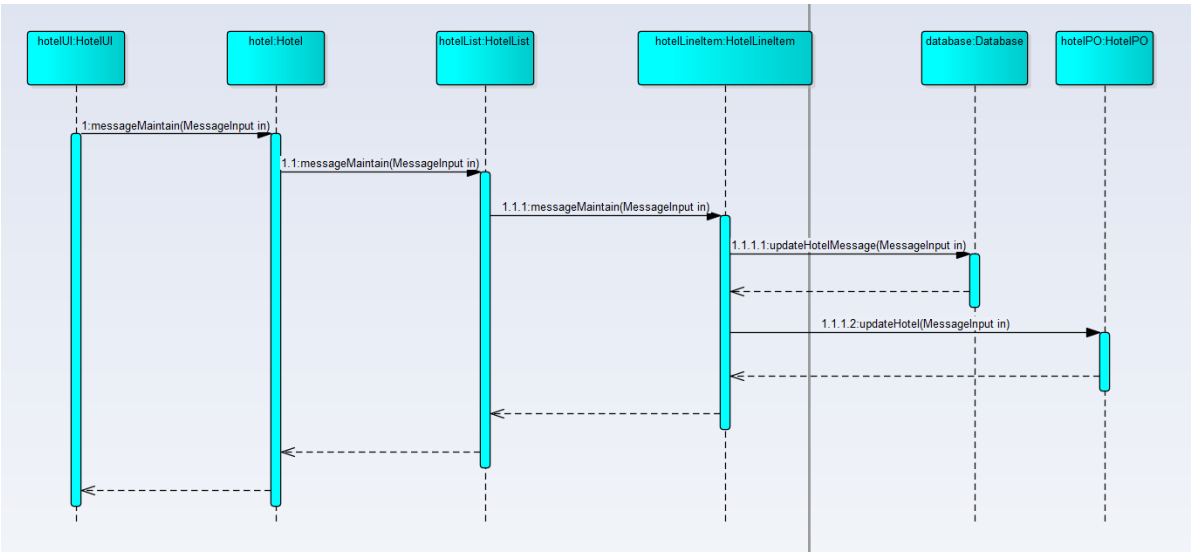

图 4-1-1-7 维护酒店基本信息的顺序图

图 4-1-1-8 表明了酒店预订系统中，在浏览酒店的过程中输入一个查找范围之后，搜索酒店业务逻辑处理的相关对象之间的协作。

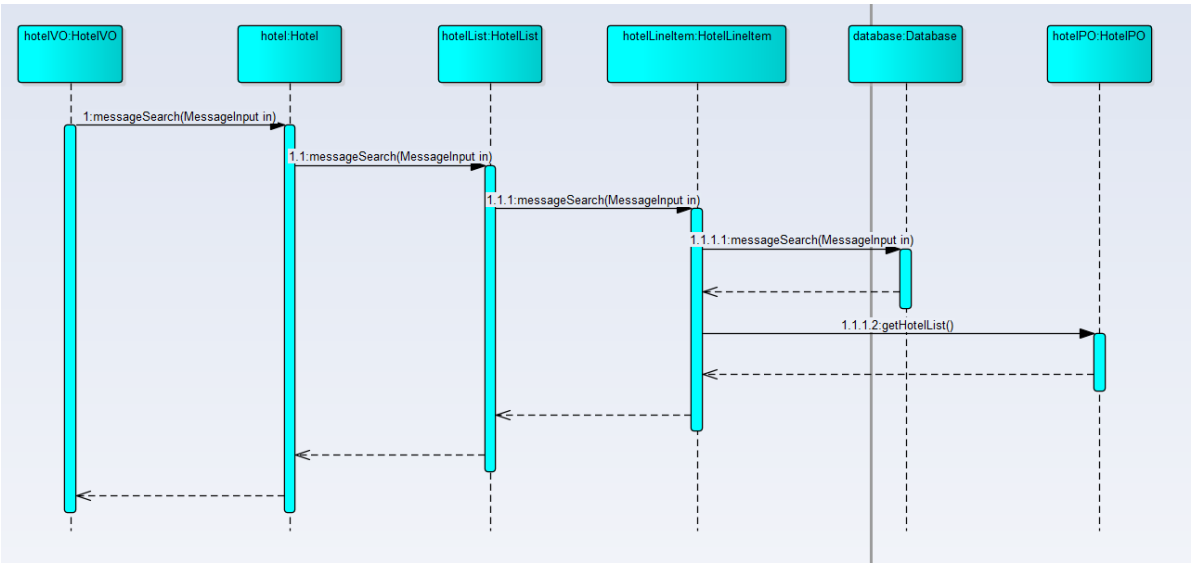

图 4-1-1-8 搜索酒店的顺序图

图 4-1-1-9 表明了酒店预订系统中，在按价格由高到低排序的过程中，酒店排序业务逻辑处理的相关对象之间的协作。

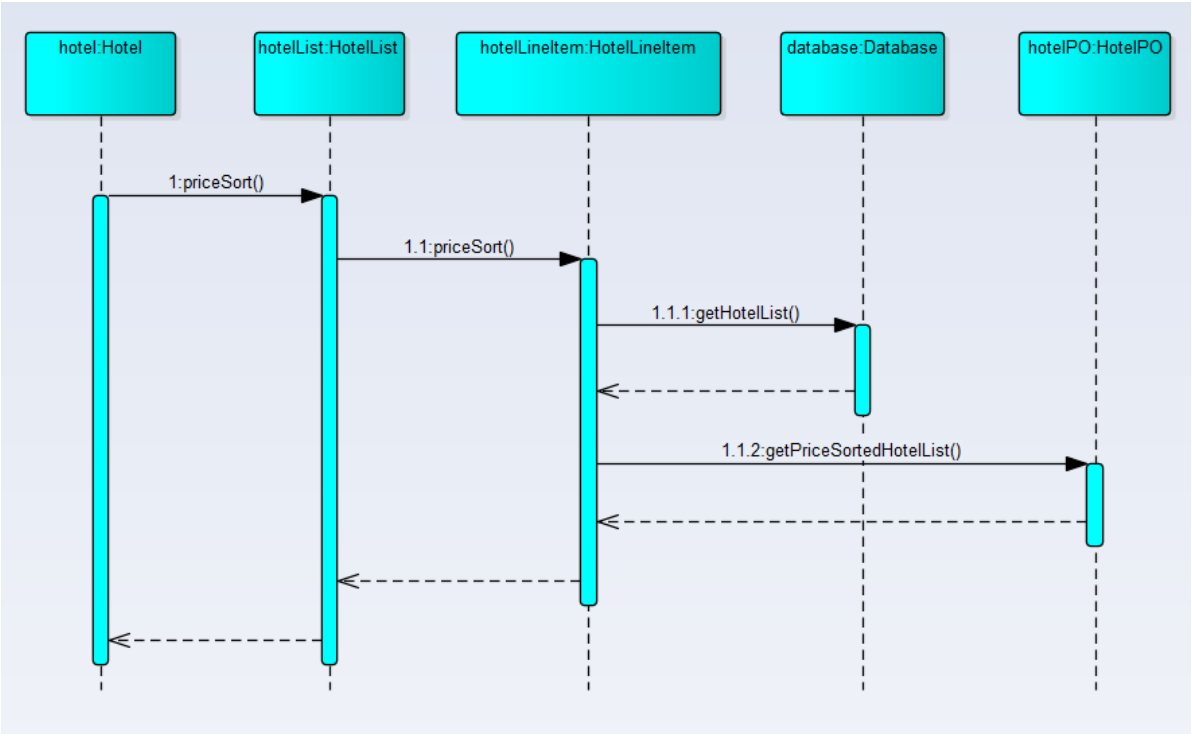

图 4-1-1-9 酒店价格由高到低排序的顺序图

图 4-1-1-10 表明了酒店预订系统中，在将酒店按分数由高到低的过程中，酒店排序业务逻辑处理的相关对象之间的协作。

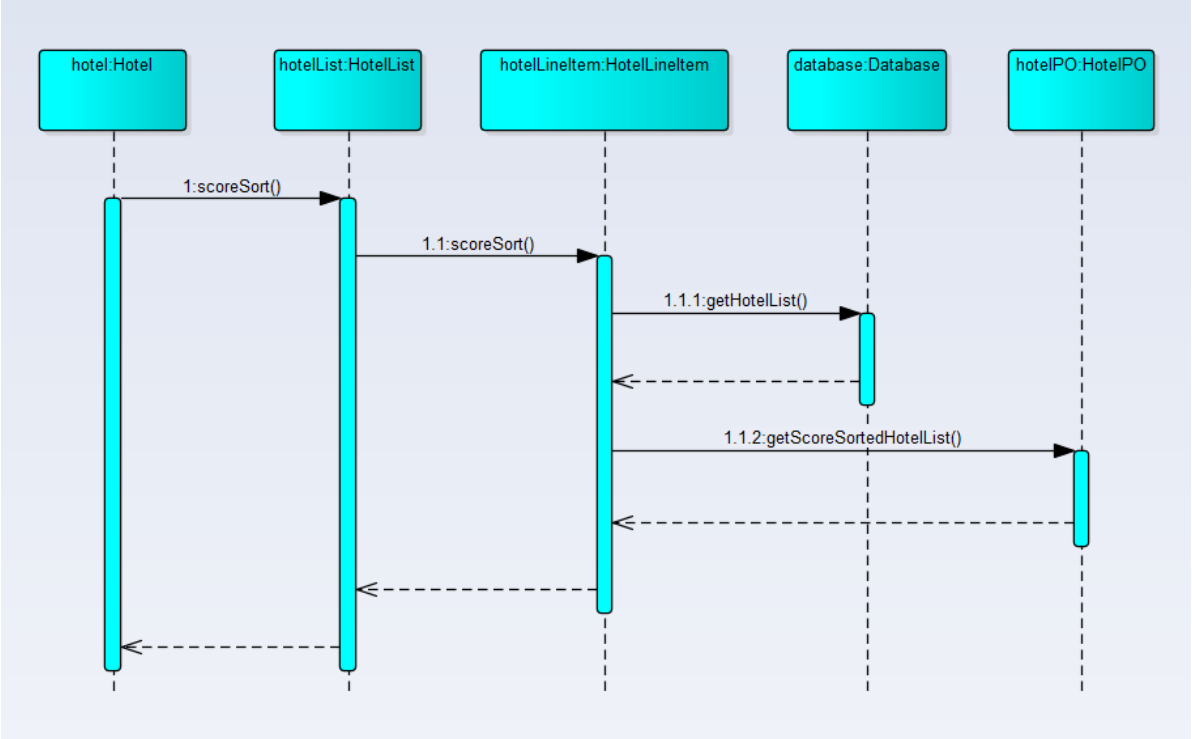

图 4-1-1-10 酒店评分由高到低排序的顺序图

图 4-1-1-11 表明了酒店预订系统中，在增加客户评价的过程中输入一个评价

之后，增加客户评价业务逻辑处理的相关对象之间的协作。

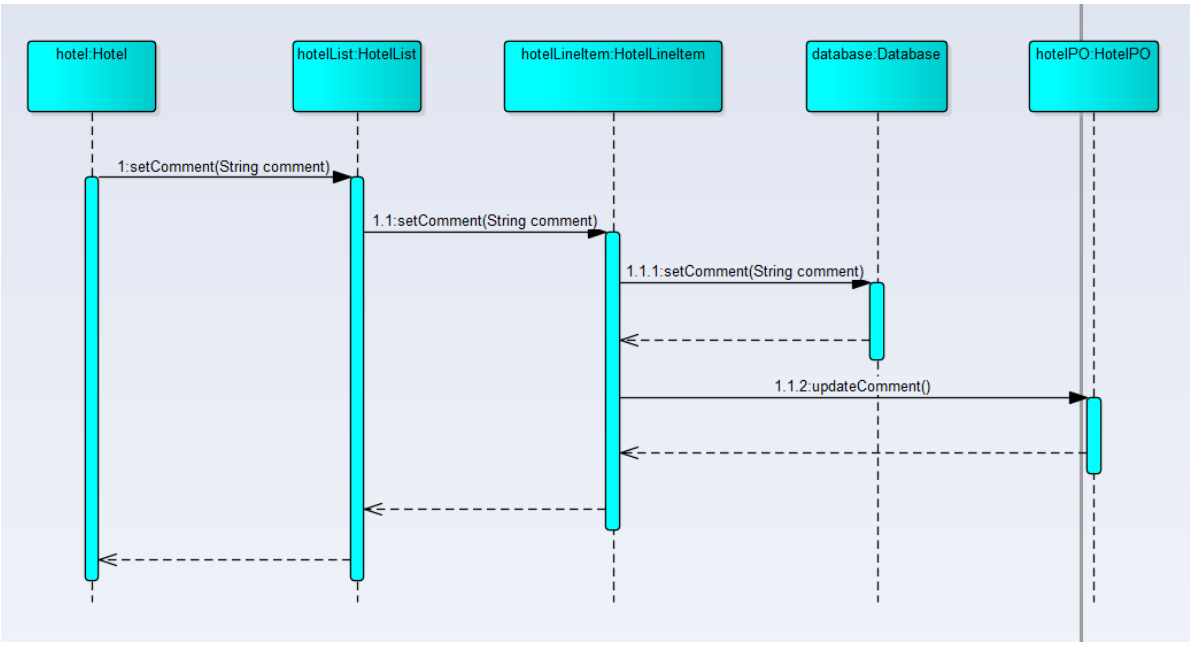

图 4-1-1-11 增加客户评价的顺序图

图 4-1-1-12 表明了酒店预订系统中，在增加评分的过程中输入一个分数之后，增加评分业务逻辑处理的相关对象之间的协作。

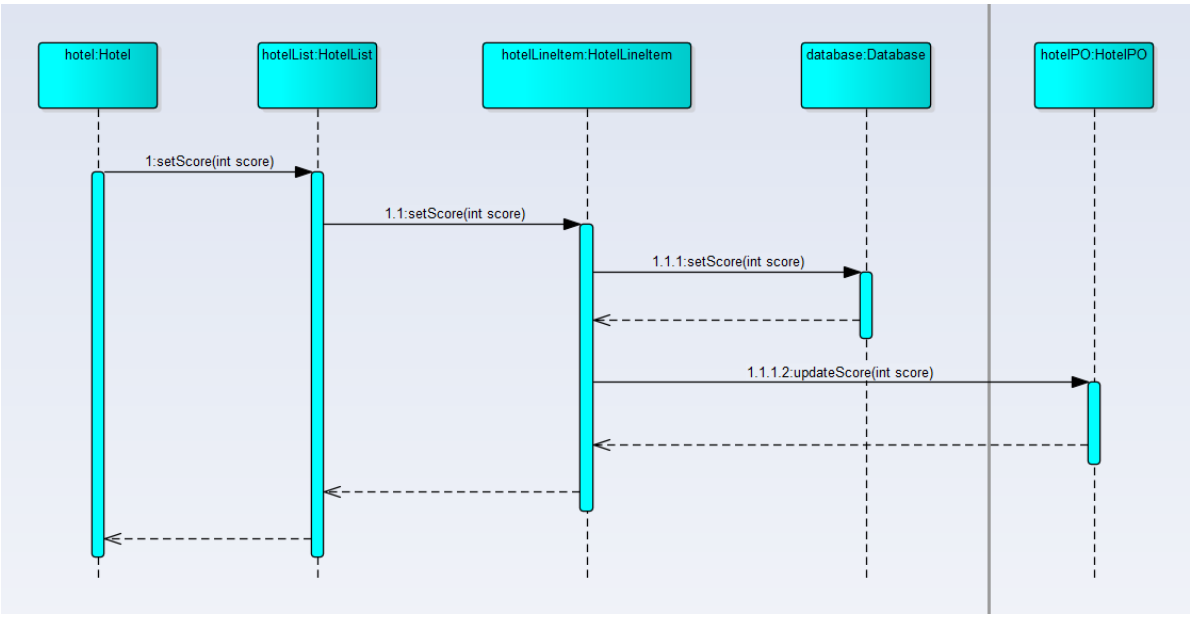

图 4-1-1-12 增加评分的顺序图

如图 4-1-1-7 所示的状态图描述了 Hotel 对象的生存期间的状态序列，引起转移的事件，以及因状态转移而伴随的动作。如果 getEverOrder 方法被 UI 调用，Hotel 进入 EverOrder 状态，之后通过 EndGetEverOrder 进入 结束状态，如果 ManagementInformation 方法或者 MaintainInformation 方法被调用，Hotel 进入 Hotel 状态，之后通过 EndUppdateInformation 进入结束状态，如果 searchInfomation 方法被调用，Hotel 进入 SearchHotelInformation 状态，之后通过 EndSearch 进入结束状态，如果 OverviewInformation 方法被调用，Hotel 进入 OverviewInformation 状态，之后通过 EndOverview 进入结束状态。

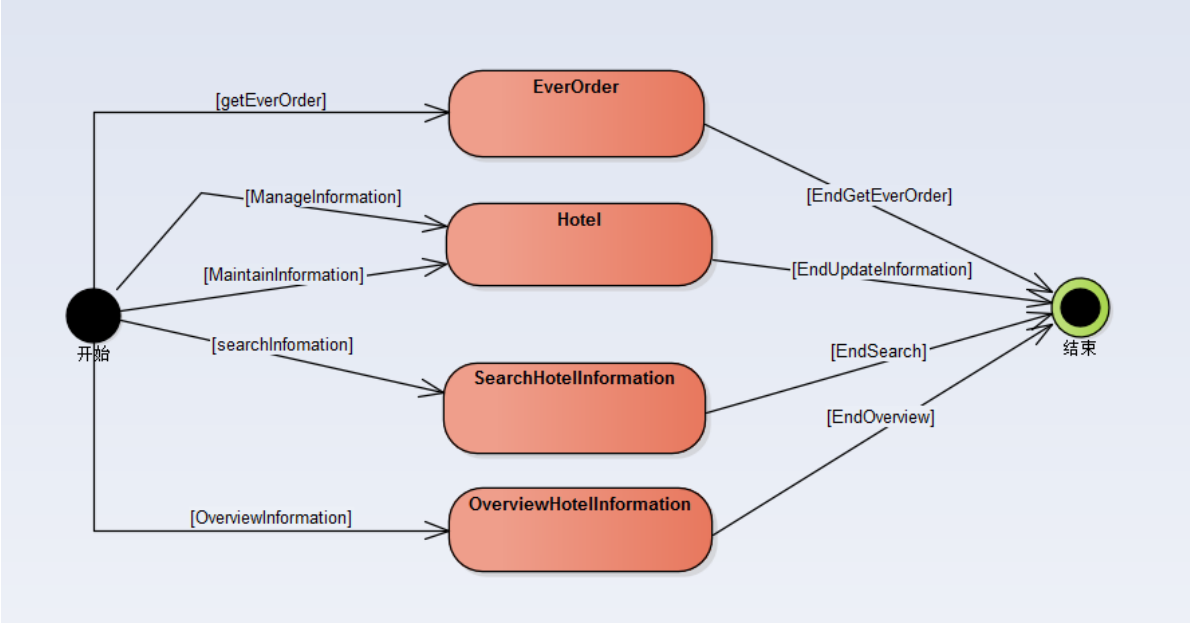

图 4-1-1-6 Hotel 对象状态图

4.1.2. roombl 模块

4.1.2.1. 模块概述

roombl 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。

roombl 模块的职责及接口参见软件系统结构设计文档。

4.1.2.2. 整体结构

根据体系结构的设计，酒店预订系统选择分层体系结构风格，将系统分为 3 层（展示层、业务逻辑层、数据层）,很好地示意了整个高层抽象。展示层包含 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。每一层之间为了增加灵活性，添加了接口，比如在展示层和业务

逻辑层之间添加 blservice.roomblservice.RoomBLService 接口，在业务逻辑层和数据层之间添加 dataservice.roomdataservice.RoomDataService 接口。RoomPO 是作为人员的持久化对象被添加到设计模型中去的。 roombl 模块的设计如图 4-1-2-1 所示。

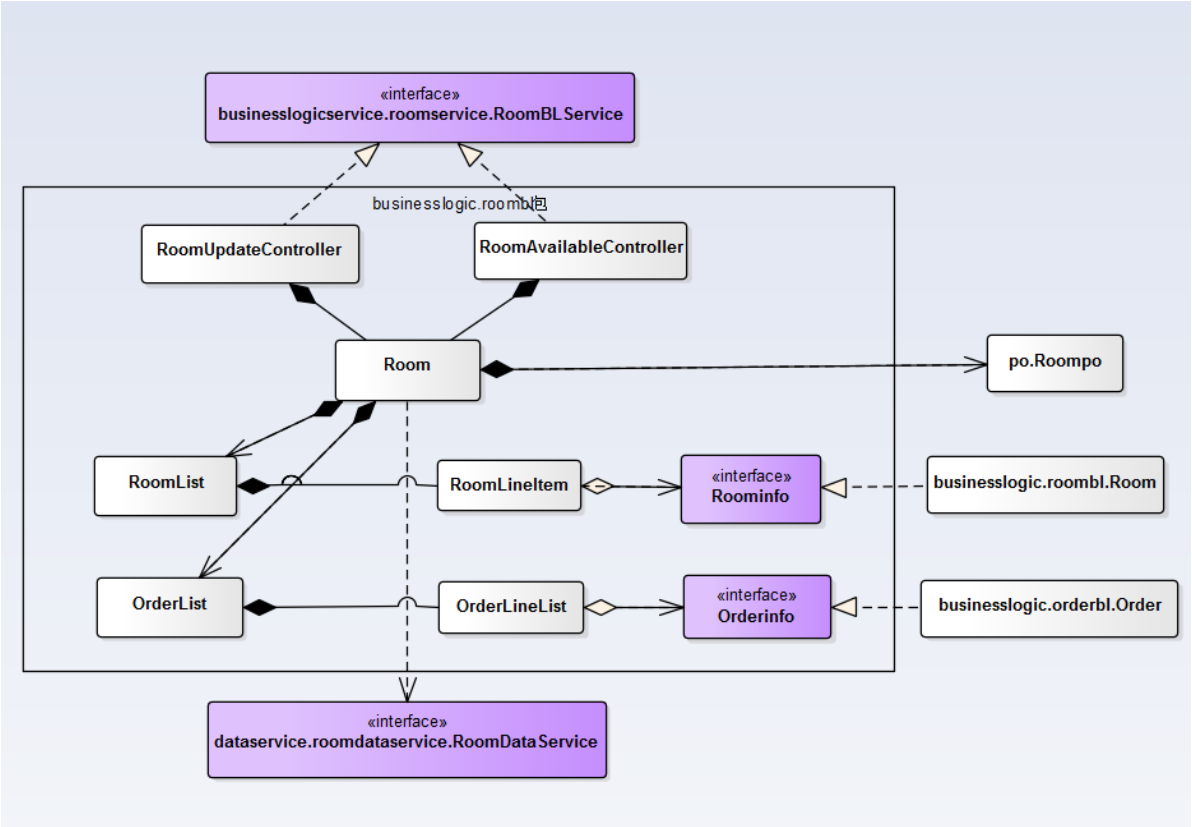

图 4-1-2-1 roombl 模块各个类的设计

roombl 模块各个类的职责如表 4-1-2-1 所示。

表 4-1-2-1 roombl 模块各个类的职责

类	职责
Room	房间的领域模型对象，拥有房间数据的房间号、房间状态、房间类型、房间数量、房间价格，可以帮助完成订房、退房的任务
RoomAvailableController	负责实现可用客房录入所需要的服务
RoomUpdateController	负责实现房间信息及状态更新所需要的服务
RoomList	获取所有房间列表
OrderList	获取所以订单列表
RoomLineItem	单个房间的信息
OrderLineItem	单个订单的信息
Roominfo	负责获得房间的对应信息的服务
Orderinfo	负责获得房间对应的订单信息的服务

4.1.2.3. 模块内部类的接口规范

RoomBL 的接口规范如表 4-1-2-2 所示。

Room 的接口规范

提供的服务（供接口）		
Room.messageupdate	语法	public ResultMessage messageupdate(MessageInput in)
	前置条件	酒店工作人员已登录成功
	后置条件	系统修改房间信息

Room.messageadd	语法	public ResultMessage messageadd(MessageInput in)
	前置条件	酒店工作人员已登录成功
	后置条件	系统修改酒店含有的房间信息
Room.getorderinfo	语法	public RoomVO messageadd(Order od)
	前置条件	客户已登录
	后置条件	得到订单上的客户的个人信息和入住退房时间

需要的服务（需接口）	
服务名	服务
DatabaseFactroy.getRoomDatabase	得到 Room 数据库的服务的引用
DatabaseFactroy.getOrderDatabase	得到 Order 数据库的服务的引用
RoomDataService.insert(RoomPO po)	插入单一持久化对象

RoomDataService.update(RoomPO po)	更新单一持久化对象
RoomDataService.delete(RoomPO po)	删除单一持久化对象
RoomDataService.changestate(String state)	在数据库更新房间状态

表 4-1-2-2 RoomBL 的接口规范

4.1.2.4. 业务逻辑层的动态模型

图 4-1-2-2 表明了酒店预订系统中，在获取订单上的客户的个人信息和入住退房时间的过程中输入一个订单 VO 之后，获取订单信息业务逻辑处理的相关对象之间的协作。

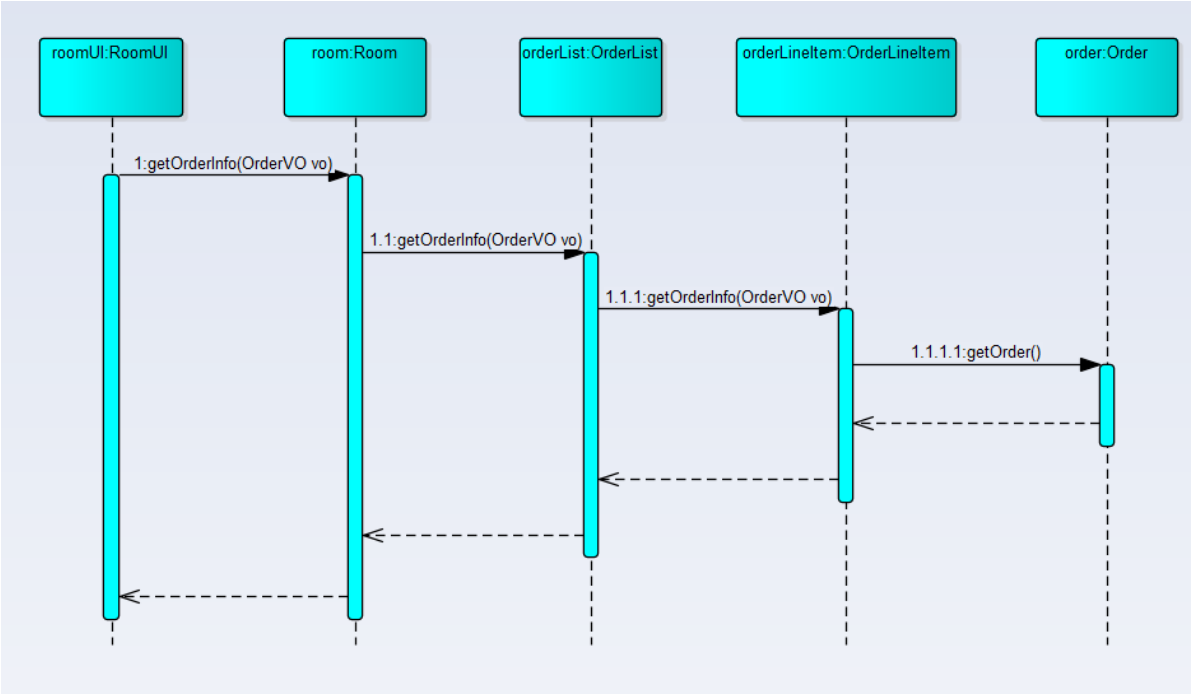

图 4-1-2-2 获取订单信息的顺序图

图 4-1-2-3 表明了酒店预订系统中，在获取某类型房间价格的过程中输入一个房间之后，获取房间价格业务逻辑处理的相关对象之间的协作。

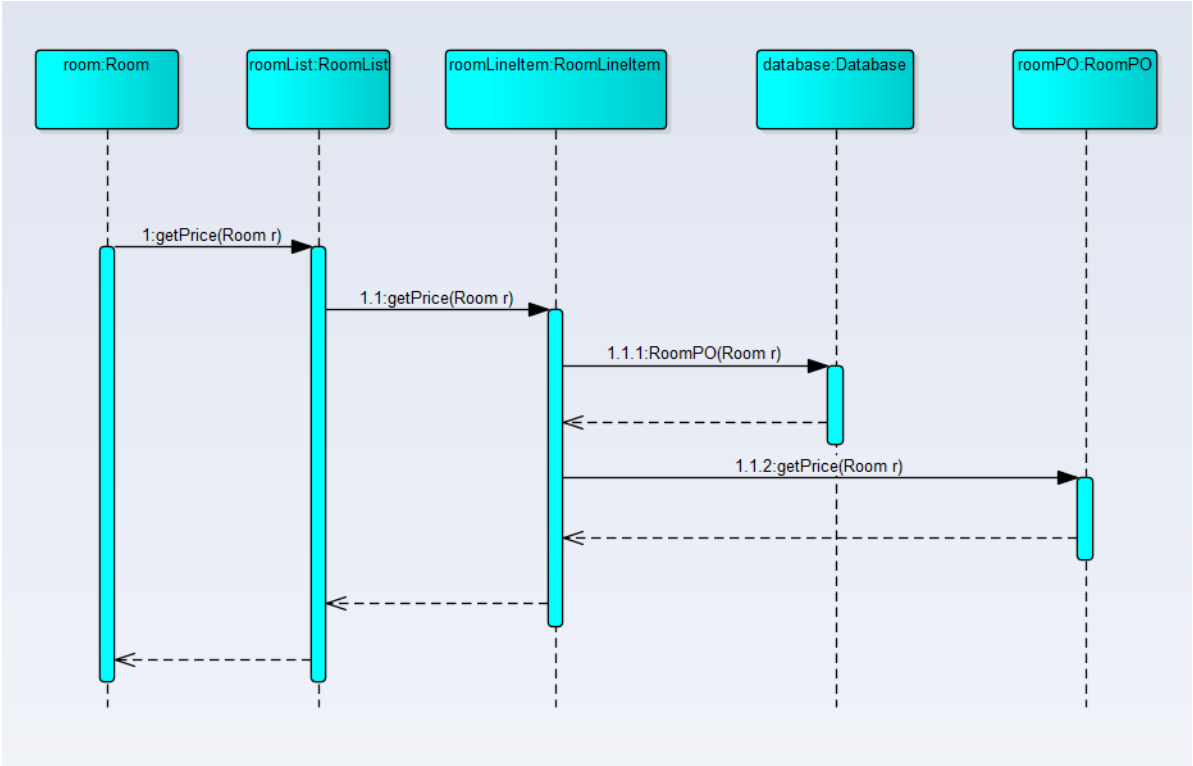

图 4-1-2-3 获取房间价格的顺序图

图 4-1-2-4 表明了酒店预订系统中，在获取可用房间的过程中，获取可用房间业务逻辑处理的相关对象之间的协作。

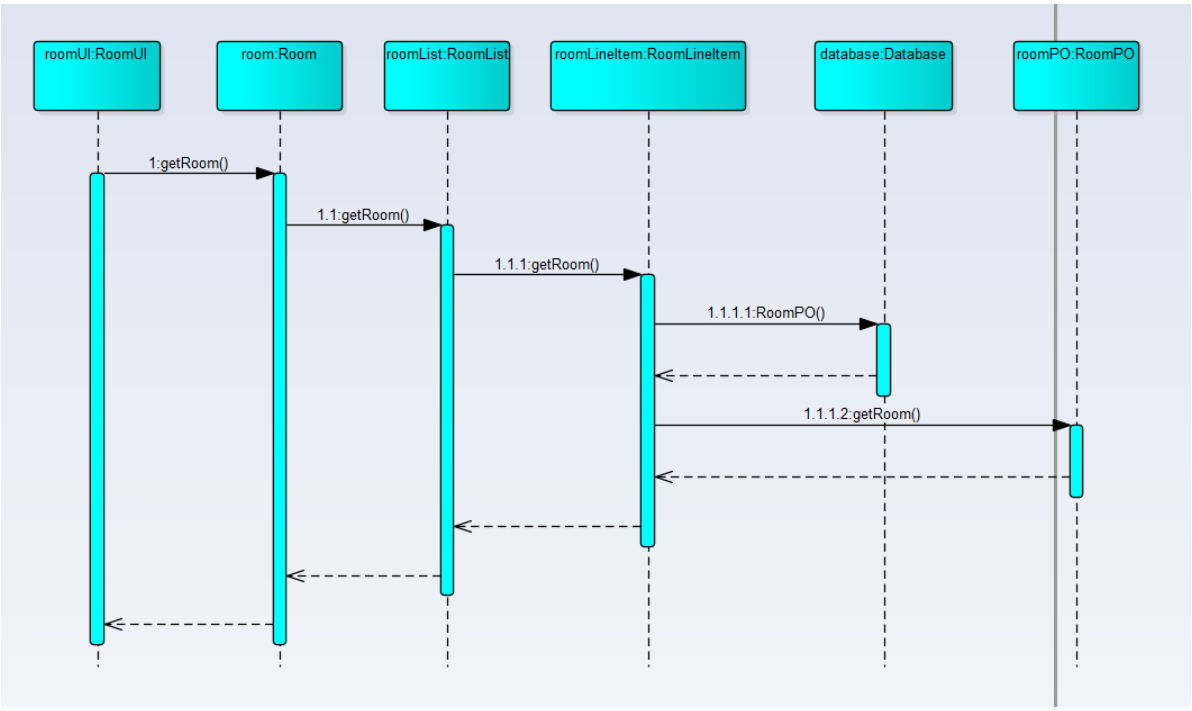

图 4-1-2-4 获取可用房间的顺序图

图 4-1-2-5 表明了酒店预订系统中，在录入可用房间的过程中输入信息之后，录入可用房间业务逻辑处理的相关对象之间的协作。

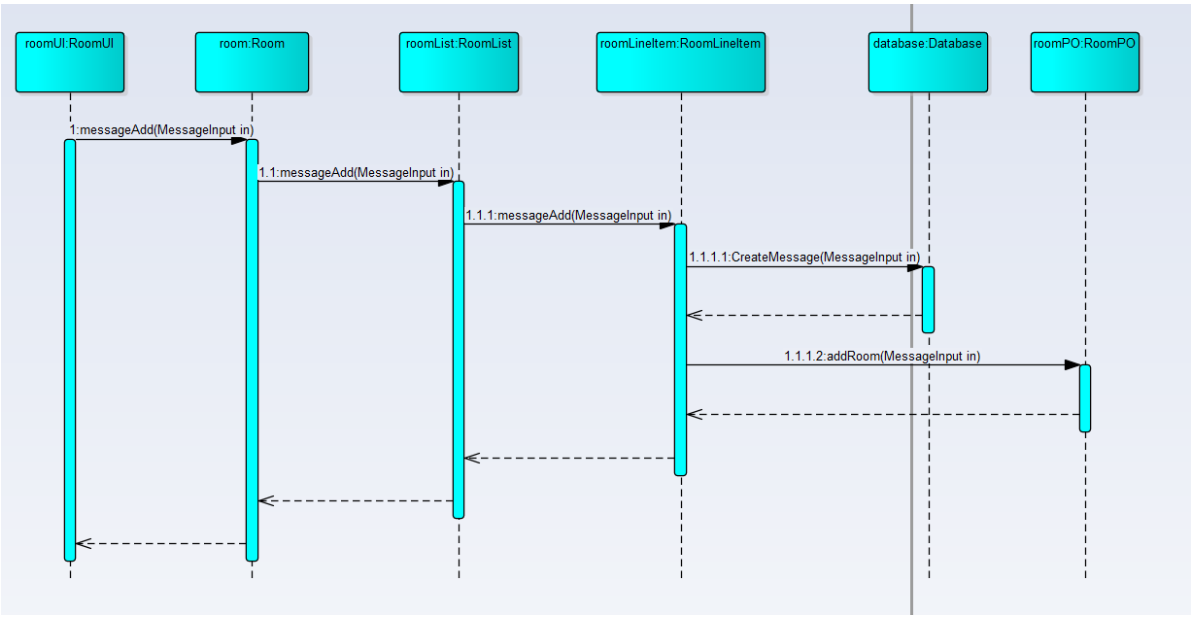

图 4-1-2-5 录入可用房间的顺序图

图 4-1-2-6 表明了酒店预订系统中，在更新房间信息的过程中输入信息

之后，更新房间信息业务逻辑处理的相关对象之间的协作。

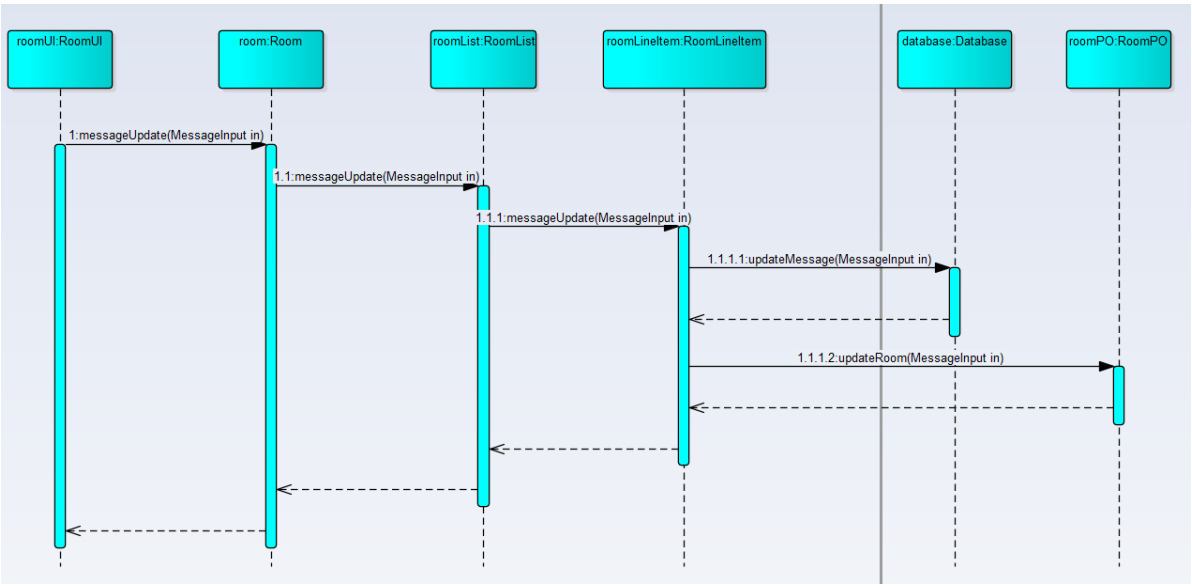

图 4-1-2-6 更新房间信息的顺序图

如图 4-1-2-7 所示的状态图描述了 Room 对象的生存期间的状态

序列，引起转移的事件，以及因状态转移而伴随的动作。如果 messageAdd 方法被 UI 调

用，Room 进入 Room 状态，之后通过 getRoom 或者通过 getPrice 进入 searchinfo 状态，最后通过 endAdd 进入结束状态，如果 RoomAvailable 方法被调用，Room 进入 InputAvailableRoom 状态，最后通过 endRoomAvailable 进入结束状态，如果 messageUpdate 方法被调用，Room 进入 updateCheckinout 状态，最后通过 endUpdate 进入结束状态 。

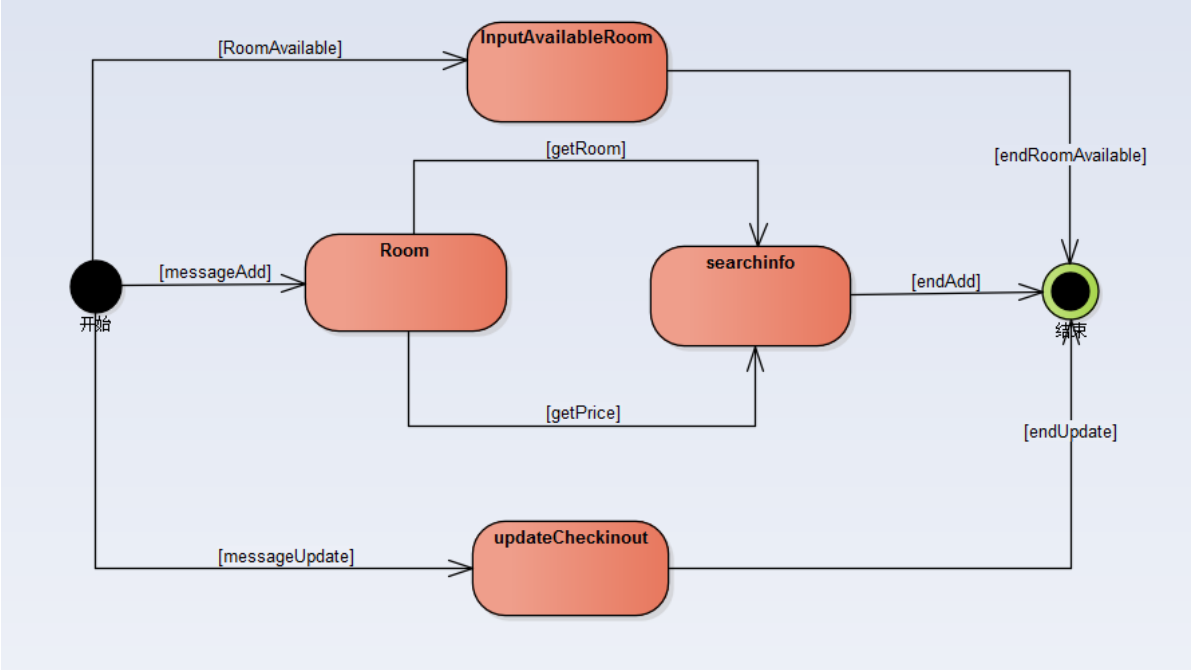

图 4-1-2-7 Room 对象状态图

4.1.3. orderbl 模块

4.1.3.1. 模块概述

orderbl 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。

orderbl 模块的职责及接口参见软件系统结构设计文档。

4.1.3.2. 整体结构

根据体系结构的设计，酒店预订系统选择分层体系结构风格，将系统分为 3 层（展示层、业务逻辑层、数据层）,很好地示意了整个高层抽象。展示层包含 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。每一层之间为了增加灵活性，添加了接口，比如在展示层和业务逻辑层之间添加 blservice.orderblservice.OrderBLService 接口，在业务逻辑层和数据层之间添加 dataservice.orderdataservice.OrderDataService 接口。OrderPO 是作为持久化对象被添加到设计模型中去的。

orderbl 模块的设计如图 4-1-3-1 所示。

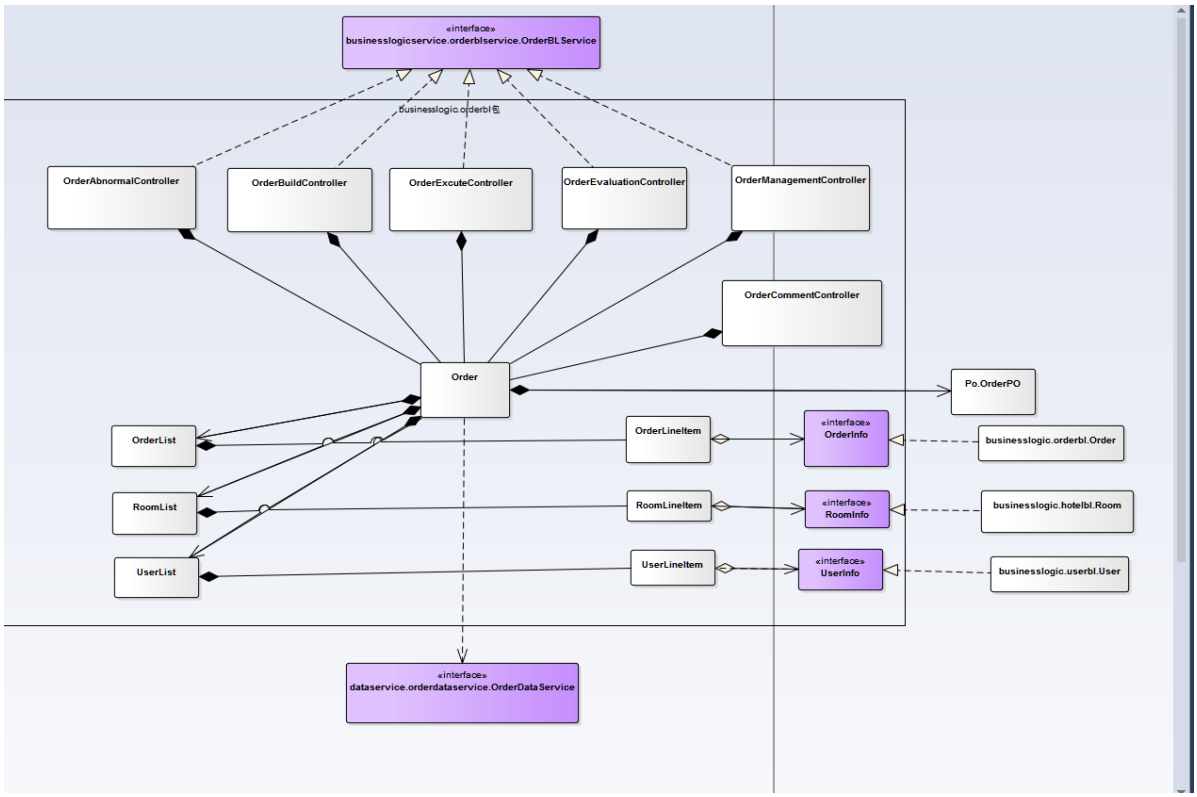

图 4-1-3-1 orderbl 模块各个类的设计

orderbl 模块各个类的职责如表 4-1-3-1 所示。

表 4-1-3-1 orderbl 模块各个类的职责

类	职责
OrderBuildController	负责实现对应于订单生成界面所需要的服务
OrderAbnormalController	负责实现对应于异常订单界面所需要的服务
OrderExecuteController	负责实现对应于订单执行界面所需要的服务
OrderEvaluationController	负责实现对应于订单评价界面所需要的服务
OrderManagementController	负责实现对应于订单管理所需要的服务
Order	订单的领域模型对象，拥有订单号，订单状态，客户名称，客户账号 ，价值，房间类型，入住时间，退房时间，最晚订单执行时间，房间数量，预计入住人数，有无儿童，信用值恢复策略，信用值，信用记

	录，会员等级，评分，评价等可以帮助完成用户信息维护所需要的服务
OrderList	封装关于 Order 的数据集合的数据结构的秘密，可以帮助完成订单管理所需要的服务
RoomList	封装关于 Room 的数据集合的数据结构的秘密，可以帮助完成房间管理所需要的服务
UserList	封装关于 User 的数据集合的数据结构的秘密，可以帮助完成用户管理所需要的服务

4.1.3.3. 模块内部类的接口规范

OrderBL 的接口规范如表 4-1-3-2 所示。

Order 的接口规范

提供的服务（供接口）		
Order.abnormalOrder	语法	public void abnormalOrder(String orderID,String userID)
	前置条件	订单异常
	后置条件	为客户减去信用值，更新会员等级
Order.whetherMake	语法	public boolean whetherMake(String uerID)
	前置条件	已提交订单

	后置条件	判断订单是否生产
Order.whetherDeduct	语法	public Boolean whetherDeduct(Date currentTime,String orderID)
	前置条件	已点击撤销订单
	后置条件	判断订单是否已撤销
Order.showDetail	语法	public OrderLineItem showDetail(String orderID)
	前置条件	已键入订单号
	后置	显示订单详情

	条件	
Order.show	语法	public List<OrderLineItem> show()
	前置条件	已登录
	后置条件	显示所有订单信息
Order.getRoomInfo	语法	public ResultMessage getRoomInfo(Order vo)
	前置条件	查看订单

	后置条件	显示订单中关于房间的信息
Order.regain	语法	public void regain(OrderVO vo,Choice choice)
	前置条件	异常订单处理成功
	后置条件	恢复客户信用值，更新会员等级
Order.makeOrder	语法	public void makeOrder(Date currentTime,Date in,Date out,Date ddl,String roomType,int num,int numOfPerson,boolean haveChild)
	前置	填写订单并提交

	条件	
	后置条件	生成一个订单对象
Order.getPrice	语法	public long getPrice(Order vo,String userID)
	前置条件	生成订单
	后置条件	计算订单价值并显示
Order.gethistory	语法	public List<OrderLineItem> gethistory(UserVO vo)

	前置条件	已登陆
	后置条件	显示用户的所有历史订单
Order.findByType	语法	public List<OrderLineItem> findByType(String type)
	前置条件	已登陆
	后置条件	根据选择的类型显示订单
Order.findByID	语法	public OrderLineItem findByID(String ID)
	前置	已登陆

	条件	
	后置条件	根据输入的 ID 显示订单
Order.findByType	语法	public List<OrderLineItem> findByType(HotelVO vo)
	前置条件	已登陆
	后置条件	根据酒店显示订单
Order.endExecute	语法	public ResultMessage endExecute()

	前置条件	完成订单
	后置条件	订单完成
Order.done	语法	public void done(String orderID,String userID)
	前置条件	已登录
	后置条件	更该订单为已执行，为客户增加信用值，更新会员等级
Order.delayin	语法	public void delayIn(String orderID,String userID)
	前置条件	订单延迟

	后置条件	更新订单类型为已执行，为客户恢复信用值，更新会员等级
Order.comment	语法	public void comment(String comment,Order order)
	前置条件	已登陆
	后置条件	更新订单信息并显示评价
Order.cancelOrder	语法	public void cancelOrder(String orderID,Date currentTime)
	前置	已登陆

	条件	
	后置条件	生成一个订单对象
Order.getPrice	语法	public long getPrice(Order vo,String userID)
	前置条件	生成订单
	后置条件	计算订单价值并显示

需要的服务（需接口）	
服务名	服务
DatabaseFactroy.getPromotionDatabase	得到 Promotion 数据库的服务的引用

DatabaseFactroy.getHotelDatabase	得到 Hotel 数据库的服务的引用
DatabaseFactroy.getOrderDatabase	得到 Order 数据库的服务的引用
DatabaseFactroy.getUserDatabase	得到 User 数据库的服务的引用
DatabaseFactroy.getRoomDatabase	得到 Room 数据库的服务的引用
OrderDataService.insert(OrderPO po)	插入单一持久化对象
OrderDataService.delete (OrderPO po)	删除单一持久化对象
OrderDataService.update(OrderPO po)	更新单一持久化对象

表 4-1-3-2 OrderBL 的接口规范

4.1.3.4. 业务逻辑层的动态模型

图 4-1-3-2 表明了酒店预订系统中，按 ID 查看订单时，查看订单业务逻辑处理的相关对象之间的协作。

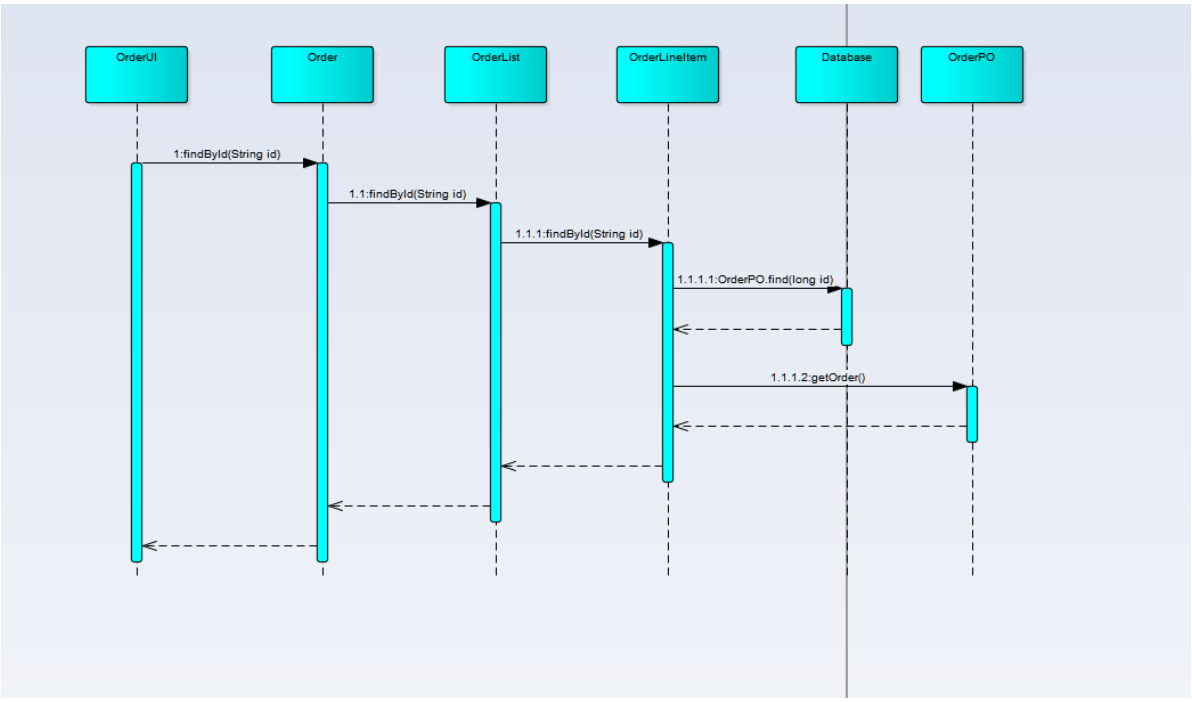

图 4-1-3-2 按 ID 查看订单的顺序图

图 4-1-3-3 表明了酒店预订系统中，在按酒店查找订单时输入一个酒店 VO 后，查找订单业务逻辑处理的相关对象之间的协作。

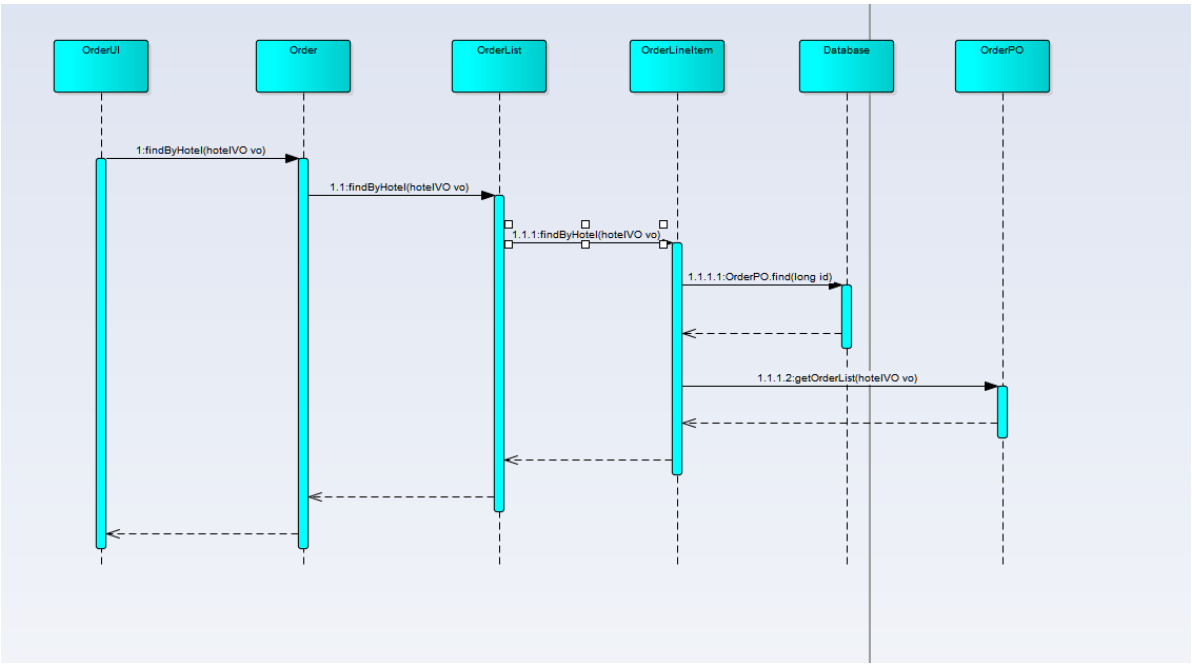

图 4-1-3-3 按酒店查找订单的顺序图

图 4-1-3-4 表明了酒店预订系统中，在按类型查找订单时输入一个类型以后，查找订单业务逻辑处理的相关对象之间的协作。

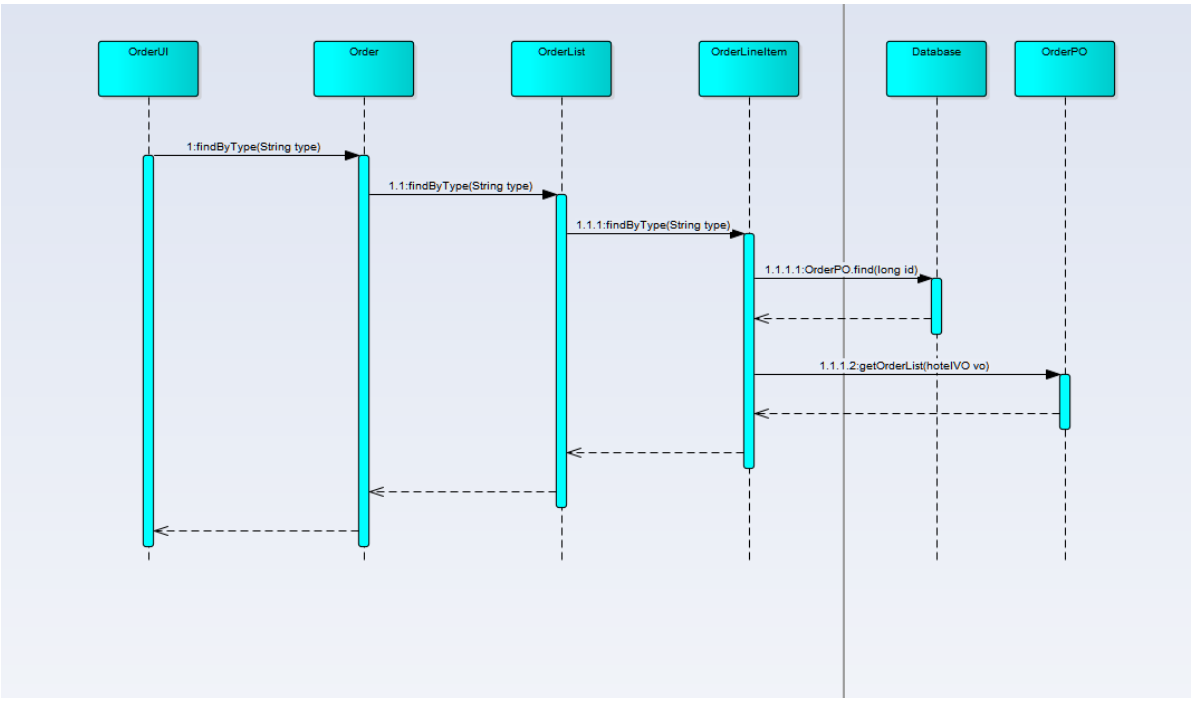

图 4-1-3-4 按类型查找订单的顺序图

图 4-1-3-5 表明了酒店预订系统中，在浏览订单时，查看订单业务逻辑处理的相关对象之间的协作。

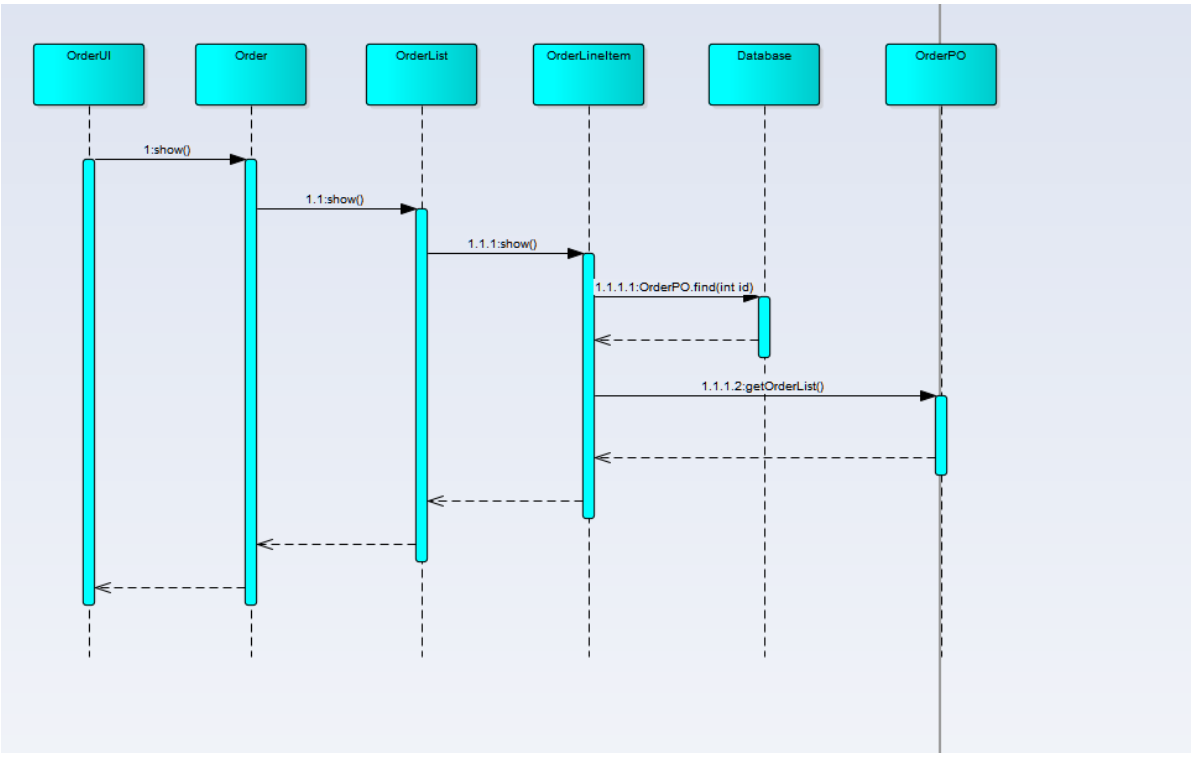

图 4-1-3-5 查看订单的顺序图

图 4-1-3-6 表明了酒店预订系统中，在浏览订单时输入一个订单 ID 后，查看订单详情业务逻辑处理的相关对象之间的协作。

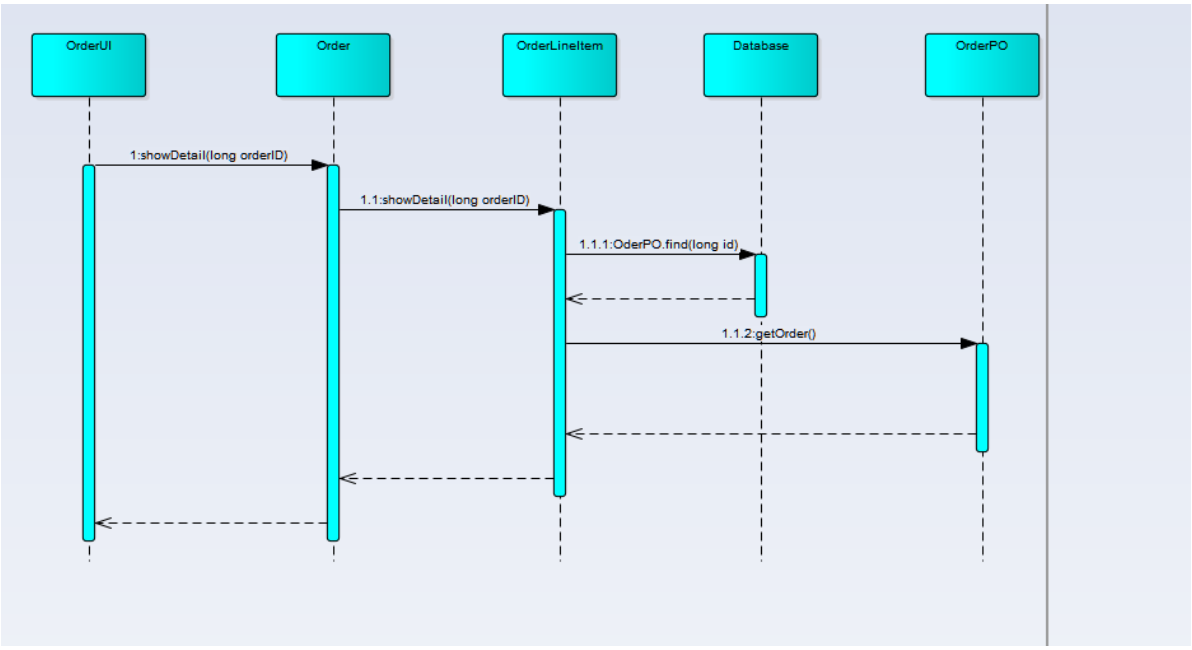

图 4-1-3-6 查看订单详情的顺序图

4-1-3-7 表明了酒店预订系统中，在撤销订单时输入一个订单编号后，撤销订单业务逻辑处理的相关对象之间的协作。

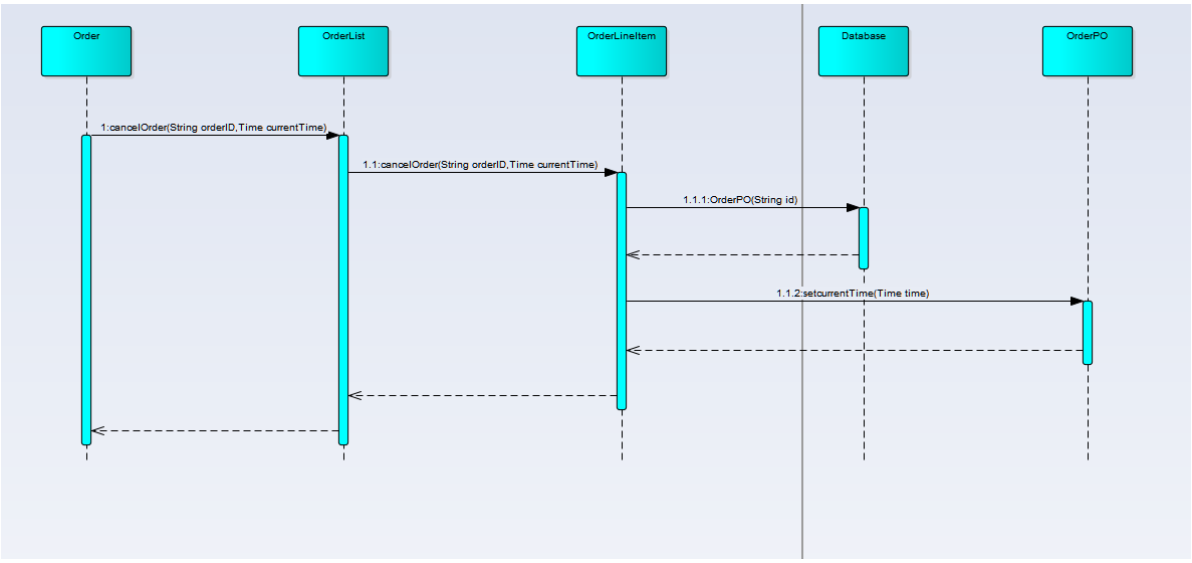

图 4-1-3-7 撤销订单的顺序图

图 4-1-3-8 表明了酒店预订系统中，在获得订单价值时输入订单 VO 和用户 ID 后，获得订单价值业务逻辑处理的相关对象之间的协作。

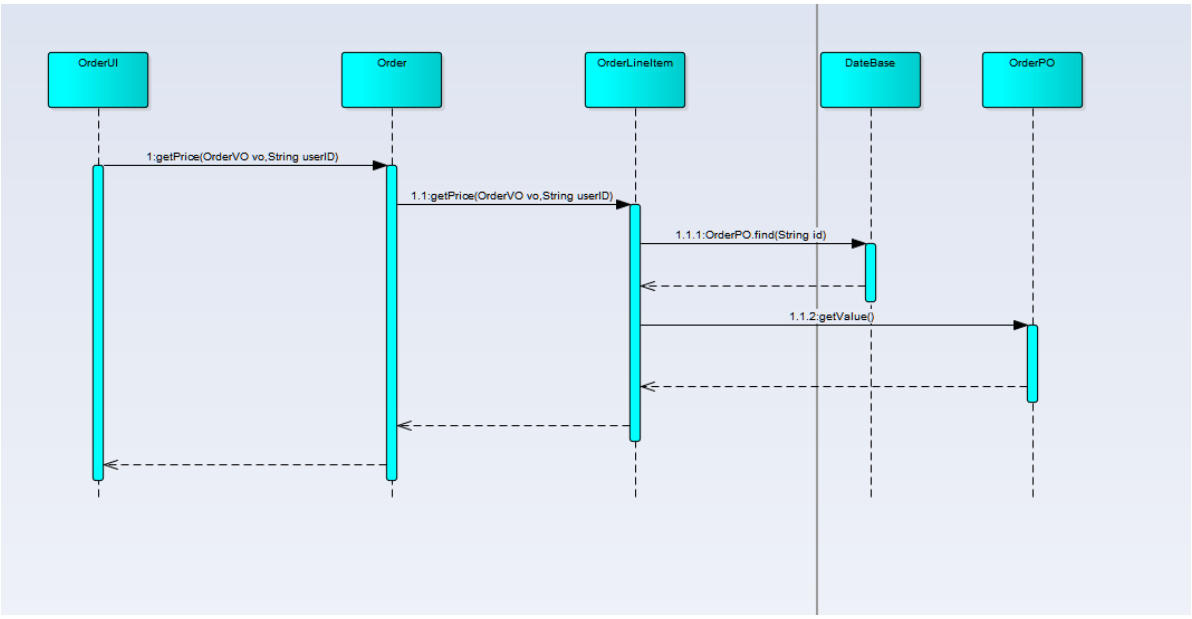

图 4-1-3-8 获取订单价值的顺序图

图 4-1-3-9 表明了酒店预订系统中，在获得更改订单的房间信息时，获得更改订单的房间信息业务逻辑处理的相关对象之间的协作。

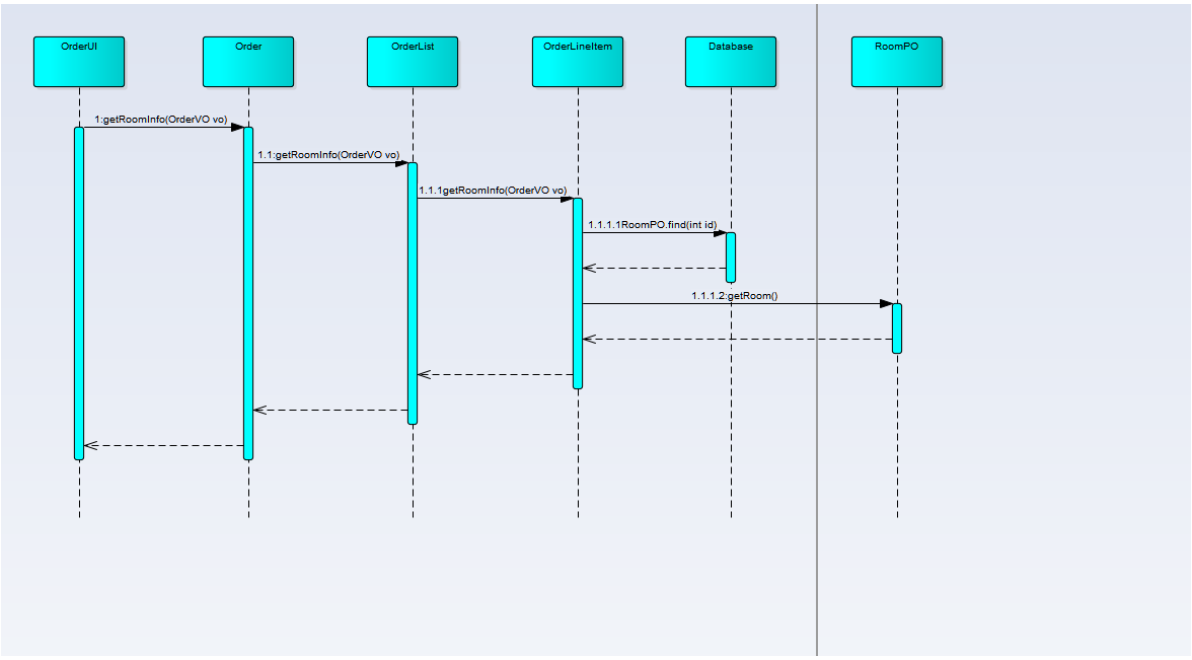

图 4-1-3-9 获得更改房间信息的订单的顺序图

图 4-1-3-10 表明了酒店预订系统中，在获得用户的历史订单时输入一个客户 VO 后，获得用户历史订单业务逻辑处理的相关对象之间的协作。

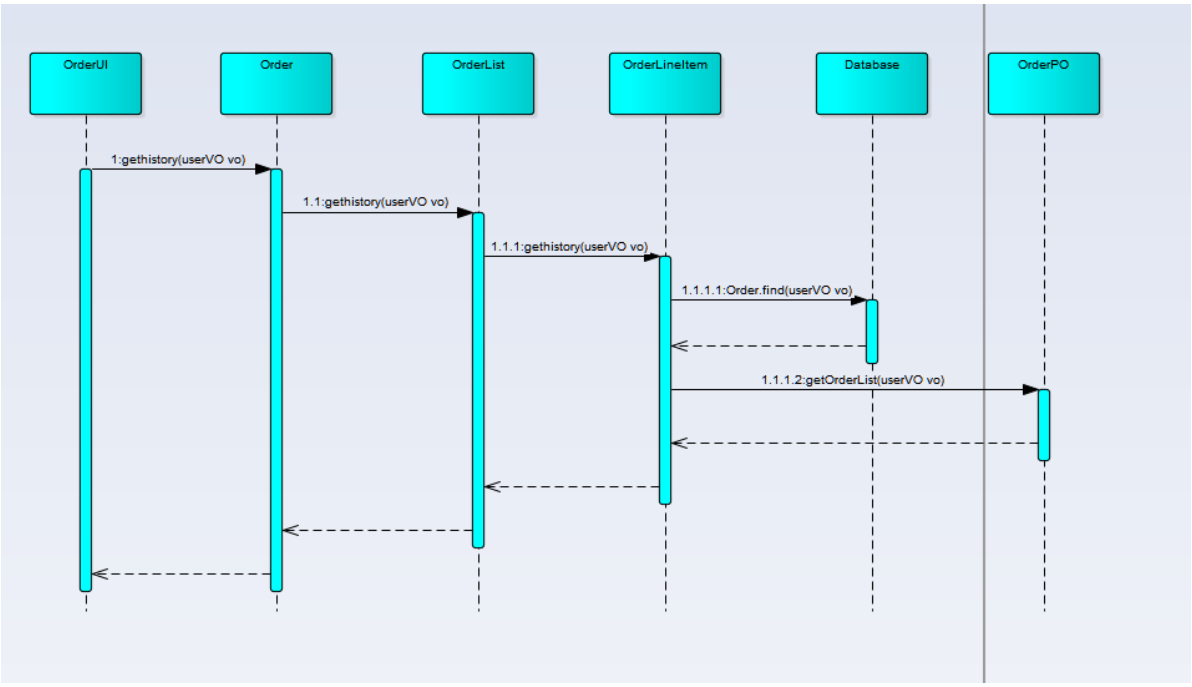

图 4-1-3-10 获得用户历史订单的顺序图

图 4-1-3-11 表明了酒店预订系统中，在结束持久化时，结束持久化业务逻辑处理的相关对象之间的协作。

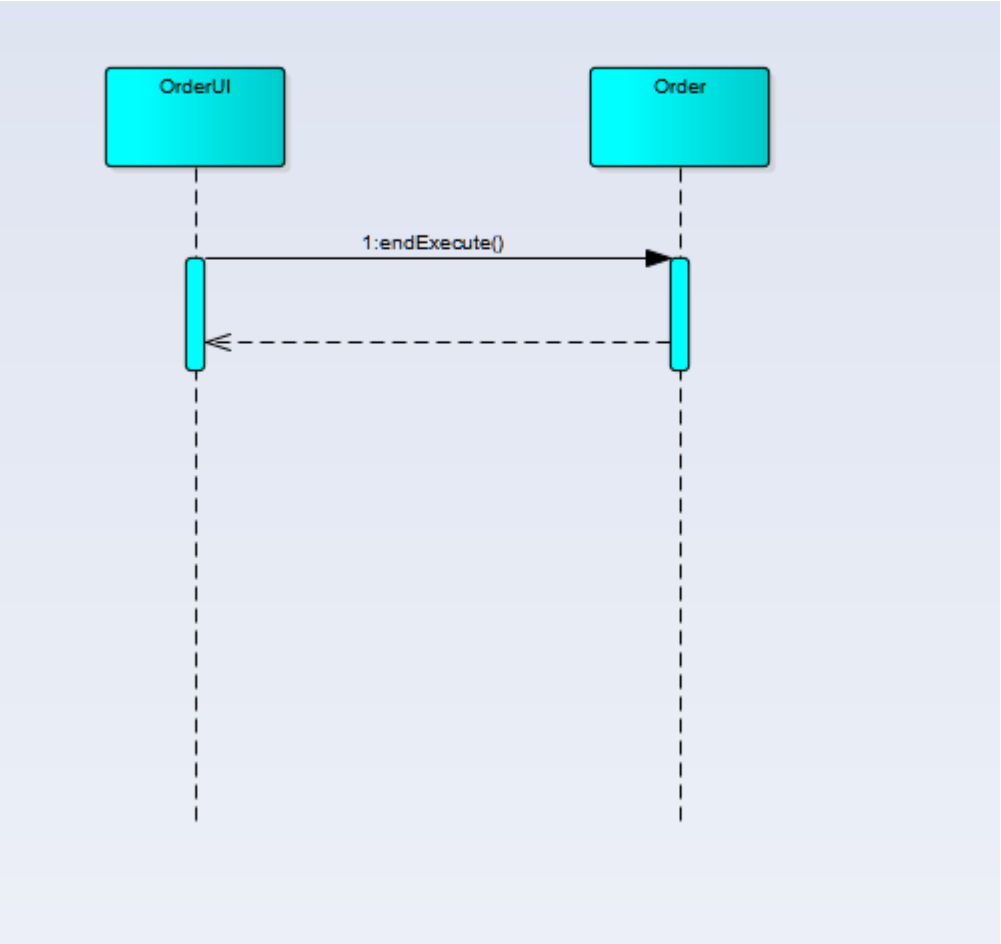

图 4-1-3-11 结束持久化的顺序图

图 4-1-3-12 表明了酒店预订系统中，在判断是否撤销成功时，判断撤销业务逻辑处理的相关对象之间的协作。

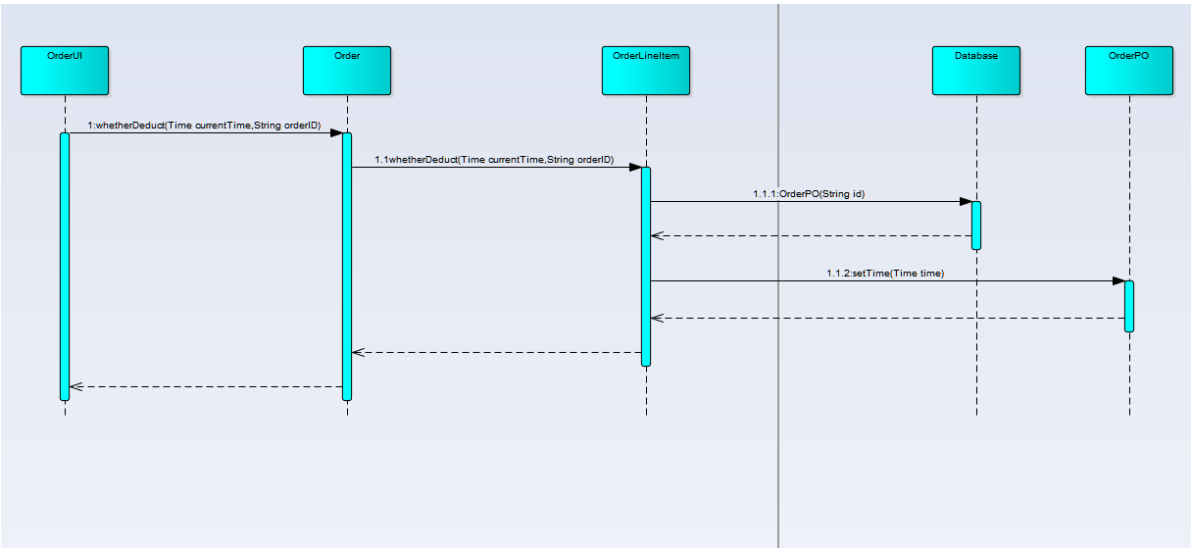

图 4-1-3-12 判断撤销是否成功的顺序图

图 4-1-3-13 表明了酒店预订系统中，在判断订单是否生成成功时输入一个用户 ID 后，判断订单生成业务逻辑处理的相关对象之间的协作。

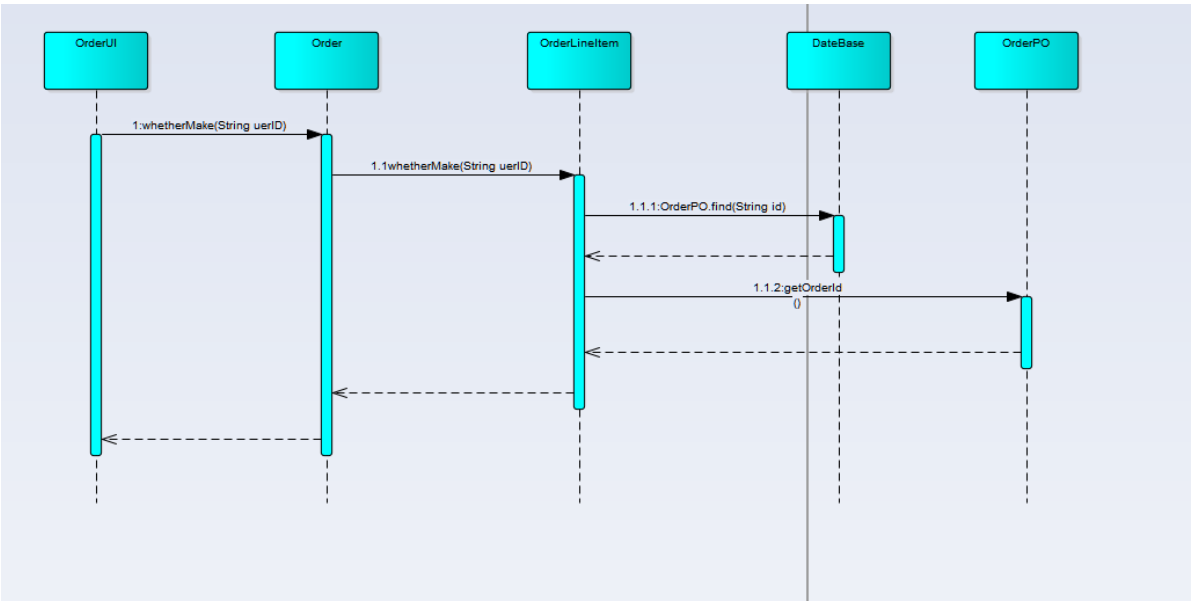

图 4-1-3-13 判断订单是否生成成功的顺序图

图 4-1-3-14 表明了酒店预订系统中，在评价订单时，评价订单业务逻辑处理的相关对象之间的协作。

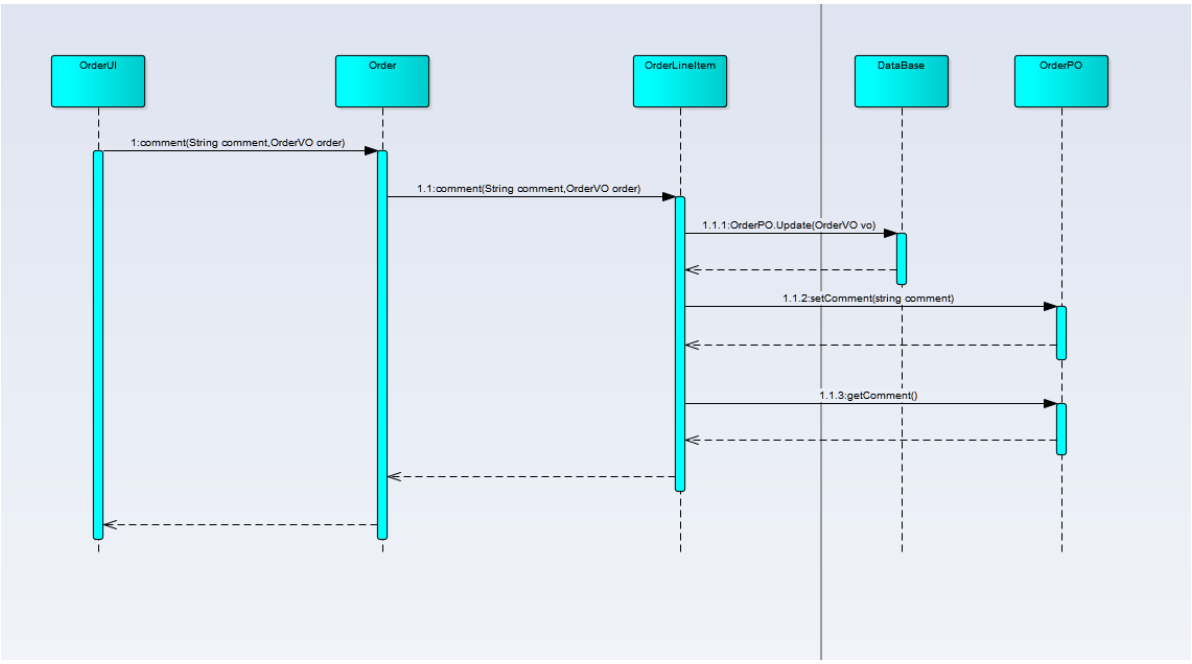

图 4-1-3-14 评价订单的顺序图

图 4-1-3-15 表明了酒店预订系统中，在取消执行订单任务时，取消执行业务逻辑处理的相关对象之间的协作。

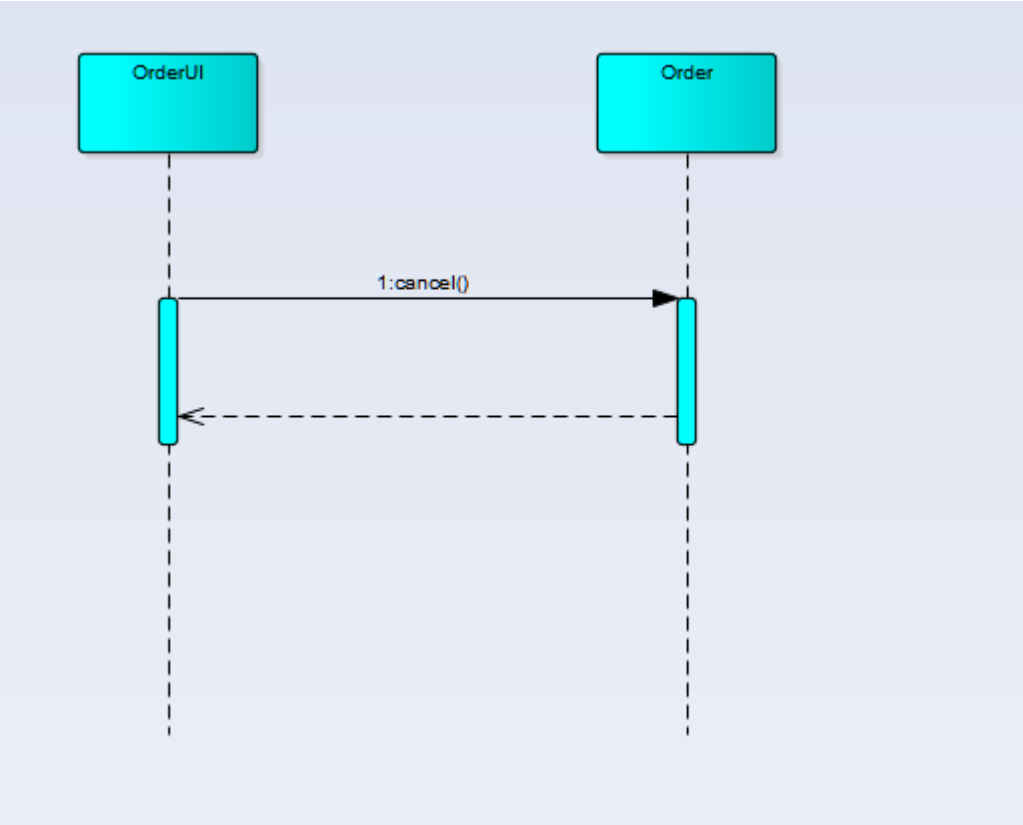

图 4-1-3-15 取消执行订单任务的顺序图

图 4-1-3-16 表明了酒店预订系统中，在申诉合理时，处理合理申诉业务逻辑处理的相关对象之间的协作。

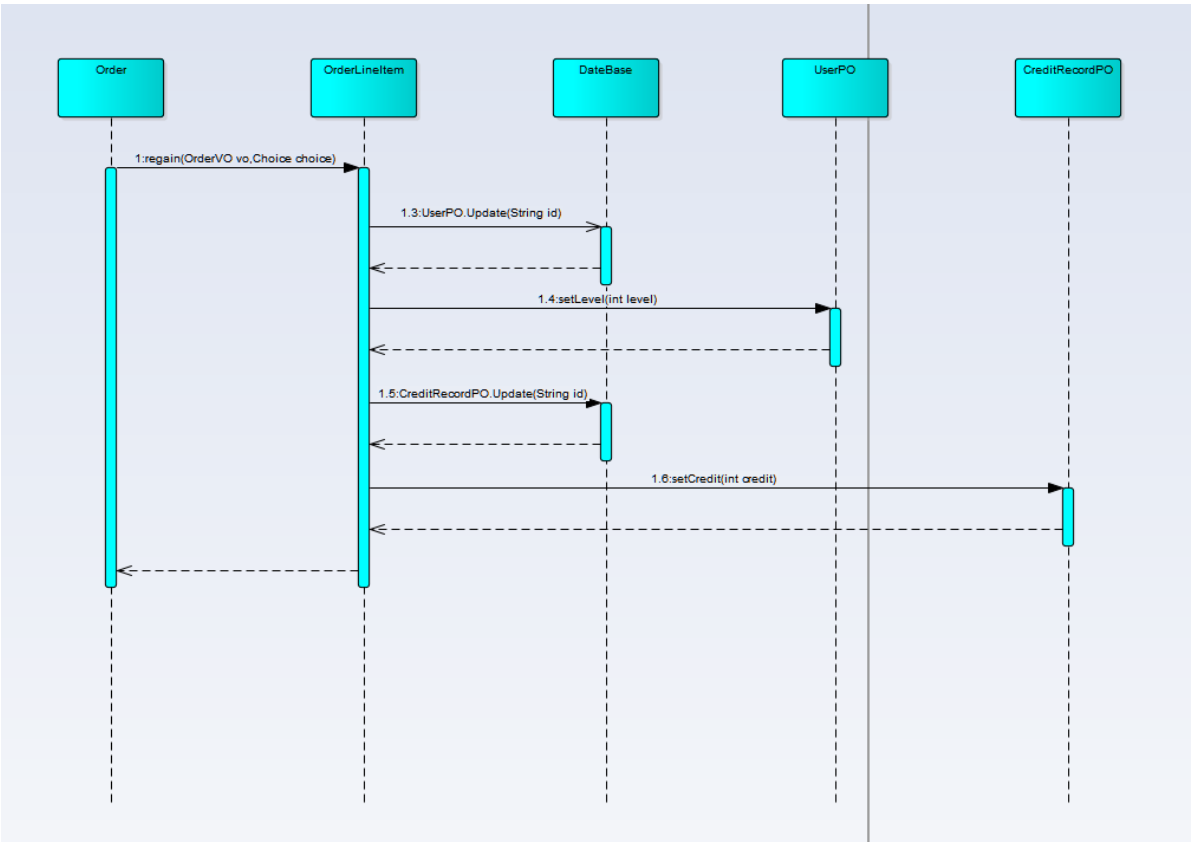

图 4-1-3-16 处理合理申诉的顺序图

图 4-1-3-17 表明了酒店预订系统中，在生成订单时，生成订单业务逻辑处理的相关对象之间的协作。

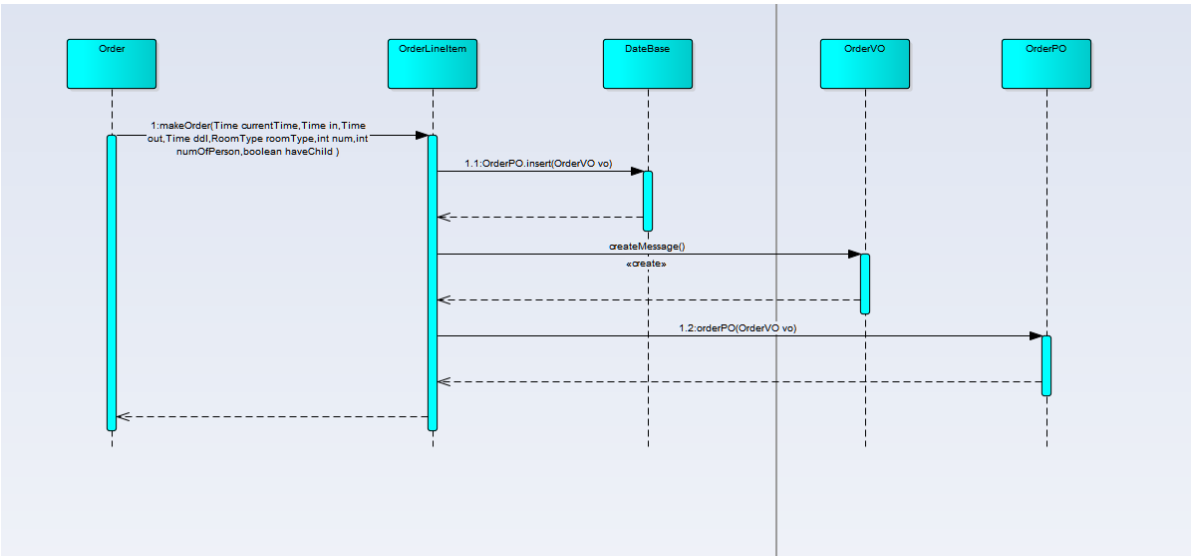

图 4-1-3-18 生成订单的顺序图

图 4-1-3-19 表明了酒店预订系统中，在支付订单时，改变订单状态业务逻辑处理的相关对象之间的协作。

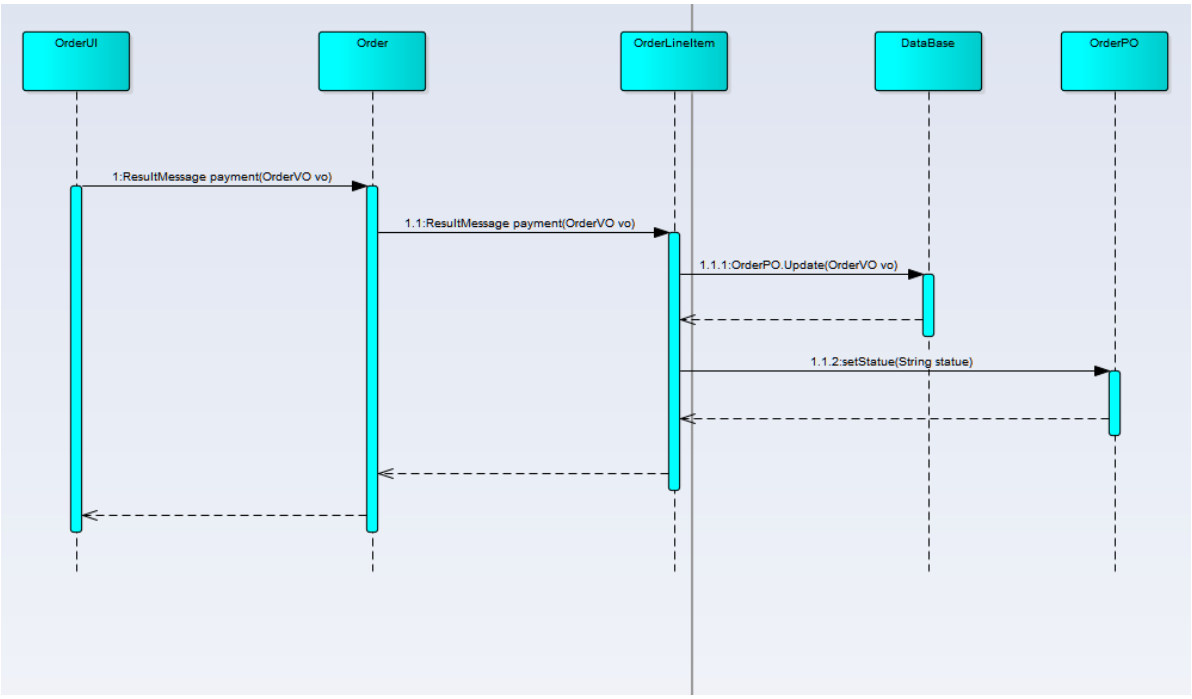

图 4-1-3-19 改变订单状态的顺序图

图 4-1-3-20 表明了酒店预订系统中，在客户延迟入住时，处理延迟订单业务逻辑处理的相关对象之间的协作。

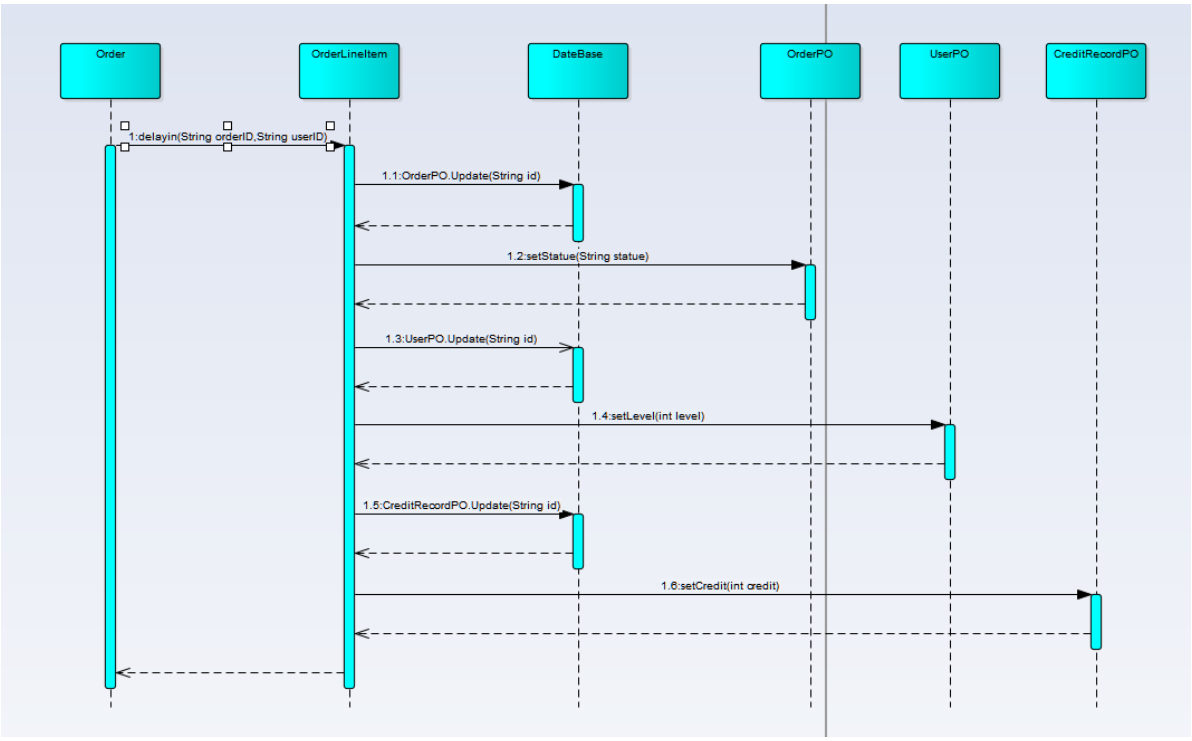

图 4-1-3-20 处理延迟订单的顺序图

图 4-1-3-21 表明了酒店预订系统中，在订单已经执行时，处理已执行订单业务逻辑处理的相关对象之间的协作。

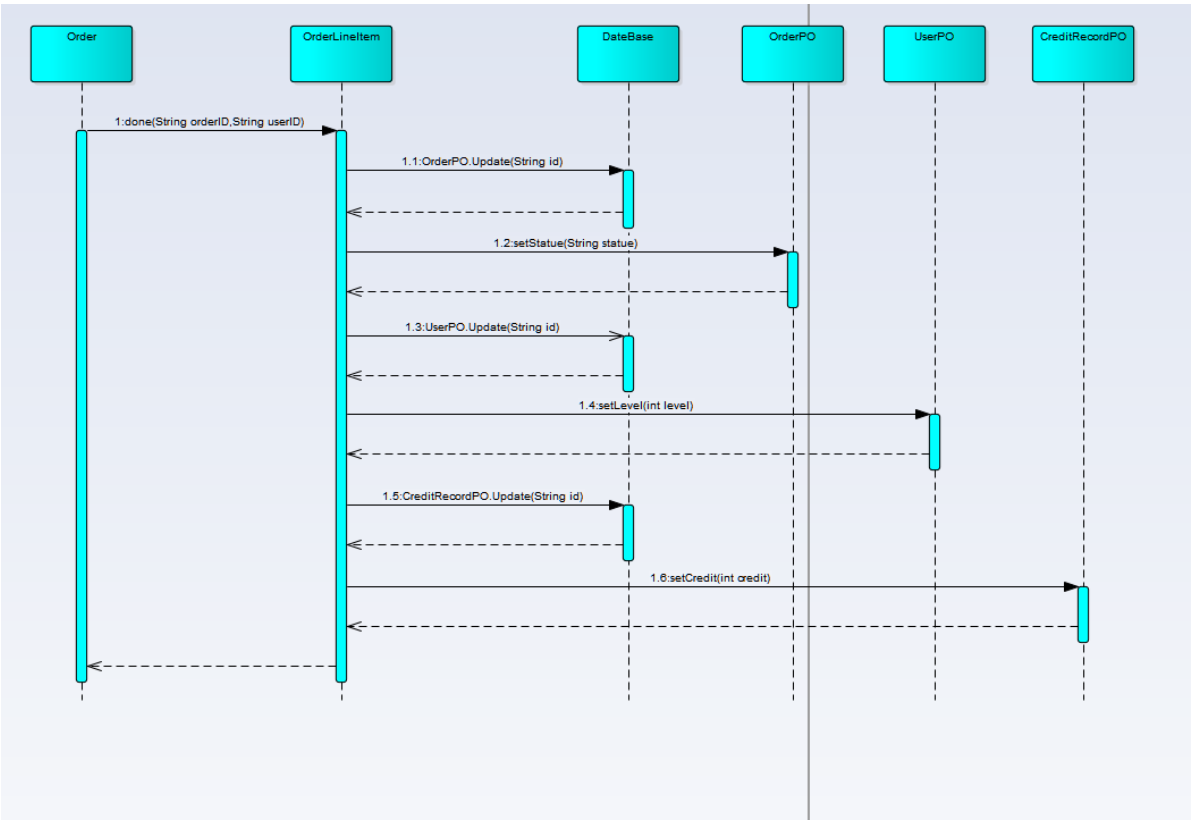

图 4-1-3-21 处理已执行订单的顺序图

图 4-1-3-22 表明了酒店预订系统中，在撤销订单时，处理撤销订单业务逻辑处理的相关对象之间的协作。

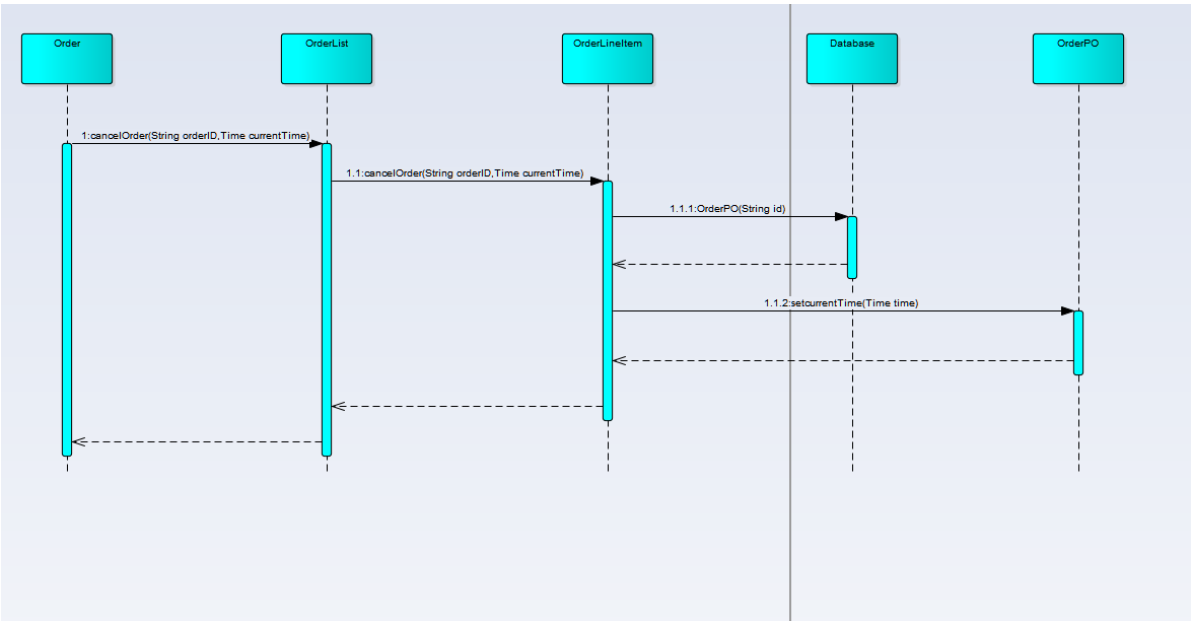

图 4-1-3-22 处理撤销订单的顺序图

图 4-1-3-23 表明了酒店预订系统中，在出现异常订单时，处理异常订单业务逻辑处理的相关对象之间的协作。

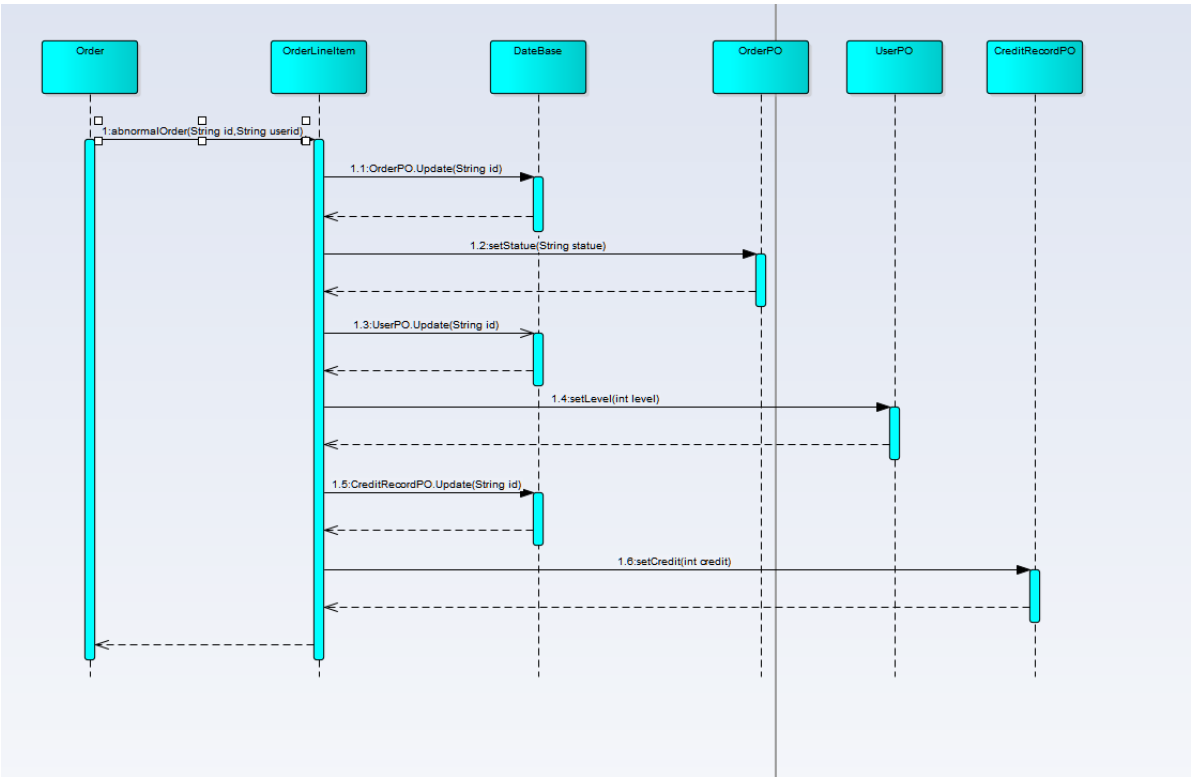

图 4-1-3-23 处理异常订单的顺序图

如图 4-1-3-3 所示的状态图描述了 Order 对象的生存期间的状态序列，引起转移的事件，以及因状态转移而伴随的动作。如果 addOrder 方法或者 browse 方法被 UI 调用，进入

Order 状态，之后通过 delete 进入 cancel 状态，最后通过 end 进入结束状态，如果 cancel 方法被调用，进入 cancel 状态，之后通过 abnormal 方法进入 abnormalOrder 状态，最后通过 end 进入结束状态，处于 Order 状态时，可通过 delay 方法进入 delayOrder 状态，最后通过 end 进入结束状态。

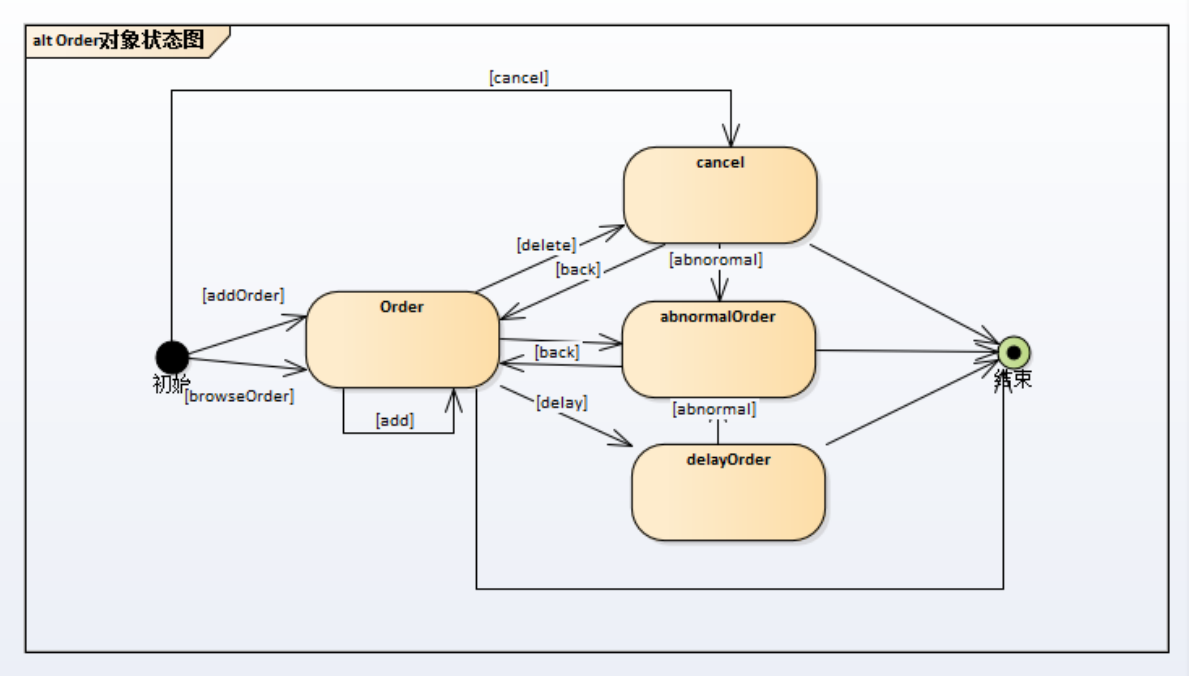

图 4-1-3-3 Order 对象状态图

4.1.4. promotionbl 模块

4.1.4.1. 模块概述

promotionbl 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。

promotionbl 模块的职责及接口参见软件系统结构设计文档。

4.1.4.2. 整体结构

根据体系结构的设计，酒店预订系统选择分层体系结构风格，将系统分为 3 层（展示层、业务逻辑层、数据层）,很好地示意了整个高层抽象。展示层包含 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。每一层之间为了增加灵活性，添加了接口，比如在展示层和业务逻辑层之间添加 blservice.promotionblservice.PromotionBLService 接口，在业务逻辑层和数据层之间添加 dataservice.promotiondataservice.PromotionDataService 接口。PromotionPO 是
作为营销策略的持久化对象被添加到设计模型中去的。

promotionbl 模块的设计如图 4-1-4-1 所示。

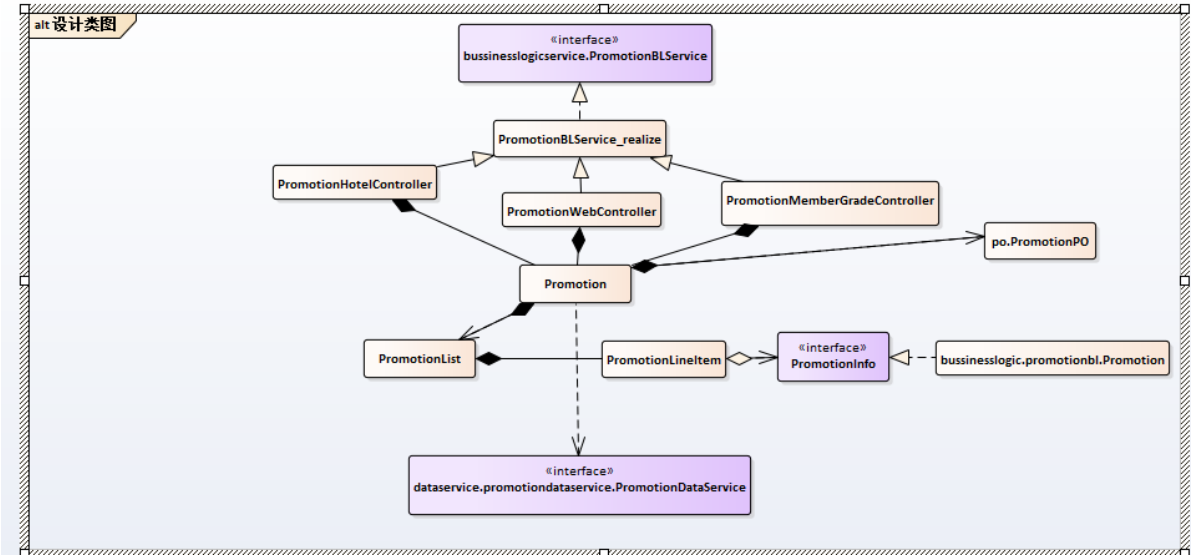

图 4-1-4-1 promotionbl 模块各个类的设计

promotionbl 模块各个类的职责如表 4-1-4-1 所示。

表 4-1-4-1 promotionbl 模块各个类的职责

类	职责
PromotionBIService_realize	负责实现与营销策略有关界面所需要的服务
PromotionHotelController	负责实现对应于酒店营销策略界面所需要的服务
PromotionWebController	负责实现对应于网站营销策略界面所需要的服务
PromotionMemberGradeController	负责实现对应于会员等级折扣界面所需要的服务
Promotion	营销策略的领域模型对象，拥有策略名称，开始时间，结束时间，会员生日，订单房间数，企业会员，会员等级，商圈，

	折扣等可以帮助完成营销策略制定所需要的服务
PromotionList	封装关于 Promotion 的数据集合的数据结构的秘密，可以帮助完成营销策略界面所需要的服务

4.1.4.3. 模块内部类的接口规范

PromotionBL 的接口规范如表 4-1-4-2 所示。

Promotion 的接口规范

提供的服务（供接口）		
Promotion.setmemberGrade	语法	public ResultMessage setmemberGrade(int creditvalue)
	前置条件	用户已登录成功
	后置条件	设置会员等级
Promotion.getPromotion	语法	public int getPromotion(String userID)
	前置条件	用户已登录成功
	后置条件	查找用户可用的营销策略数
Promotion.add	语法	public int add(String promotionnumber,String promotionid,Time begintime,Time endtime,Time birthday,int roomnumber,boolean corporatemember,String hoteldistrict,ResultMessage membergrade,doubledcount)
	前置条件	用户已登录成功
	后置条件	添加营销策略
Promotion.delete	语法	public int delete(Promotion promotion)
	前置条件	酒店工作人员已登陆
	后置条件	删除营销策略

需要的服务（需接口）	
服务名	服务

DatabaseFactroy.getPromotionDatabase	得到 Promotion 数据库的服务的引用
DatabaseFactroy.getHotelDatabase	得到 Hotel 数据库的服务的引用
DatabaseFactroy.getOrderDatabase	得到 Order 数据库的服务的引用
DatabaseFactroy.getUserDatabase	得到 User 数据库的服务的引用
PromotionDataService.insert(PromotionPO po)	插入单一持久化对象
PromotionDataService.delete (PromotionPO po)	删除单一持久化对象
PromotionDataService.memberlevelinsert(MessageInput in)	在数据区添加会员等级制度
PromotionDataService.memberlevelupdate(MessageInput in)	在数据区更新会员等级制度

表 4-1-4-2 PromotionBL 的接口规范

4.1.4.4. 业务逻辑层的动态模型

图 4-1-4-2 表明了酒店预订系统中，在查看营销策略时，查看营销策略业务逻辑处理的相关对象之间的协作。

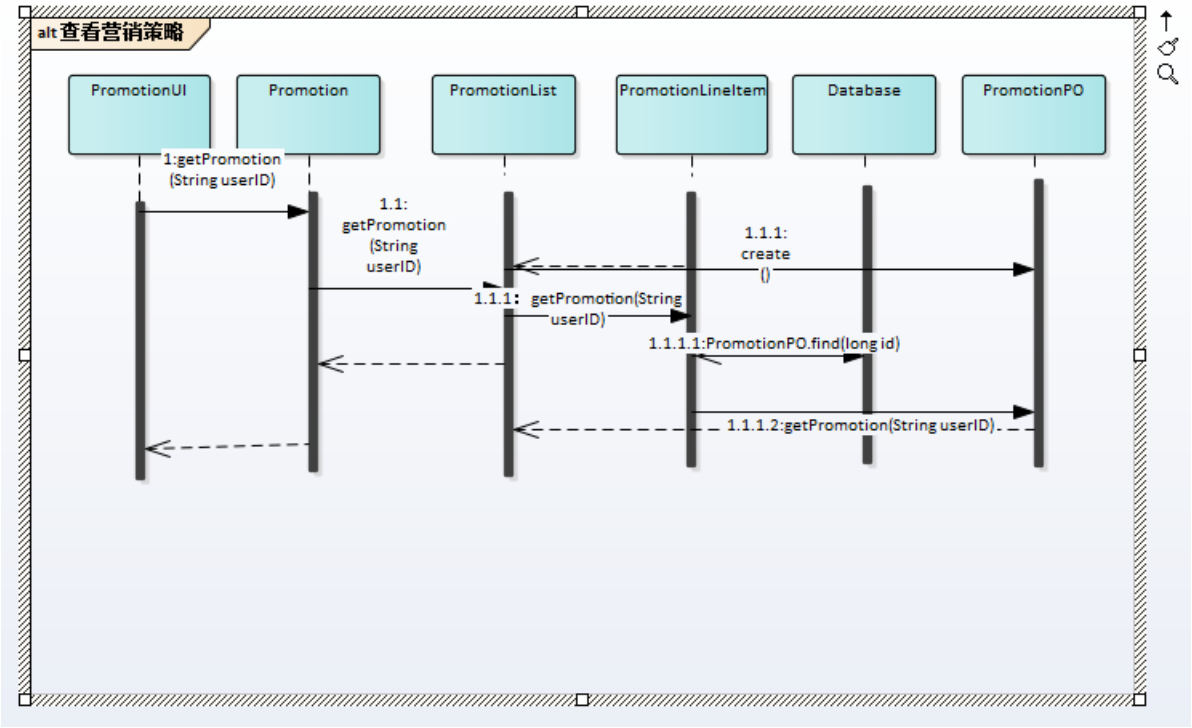

图 4-1-4-2 查看营销策略的顺序图

图 4-1-4-3 表明了酒店预订系统中，在制定会员等级折扣时，制定会员等级业务逻辑处理的相关对象之间的协作。

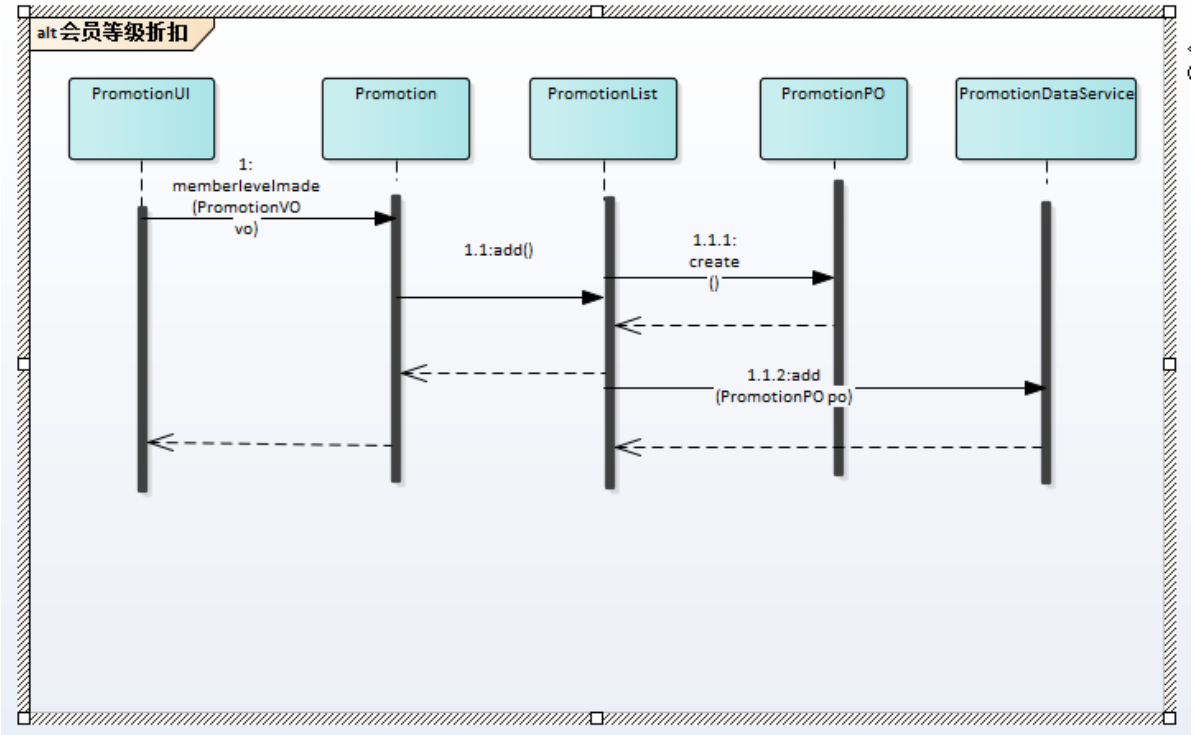

图 4-1-4-3 制定会员等级折扣的顺序图

图 4-1-4-4 表明了酒店预订系统中，在制定酒店营销策略时，制定酒店营销策略业务逻辑处理的相关对象之间的协作。

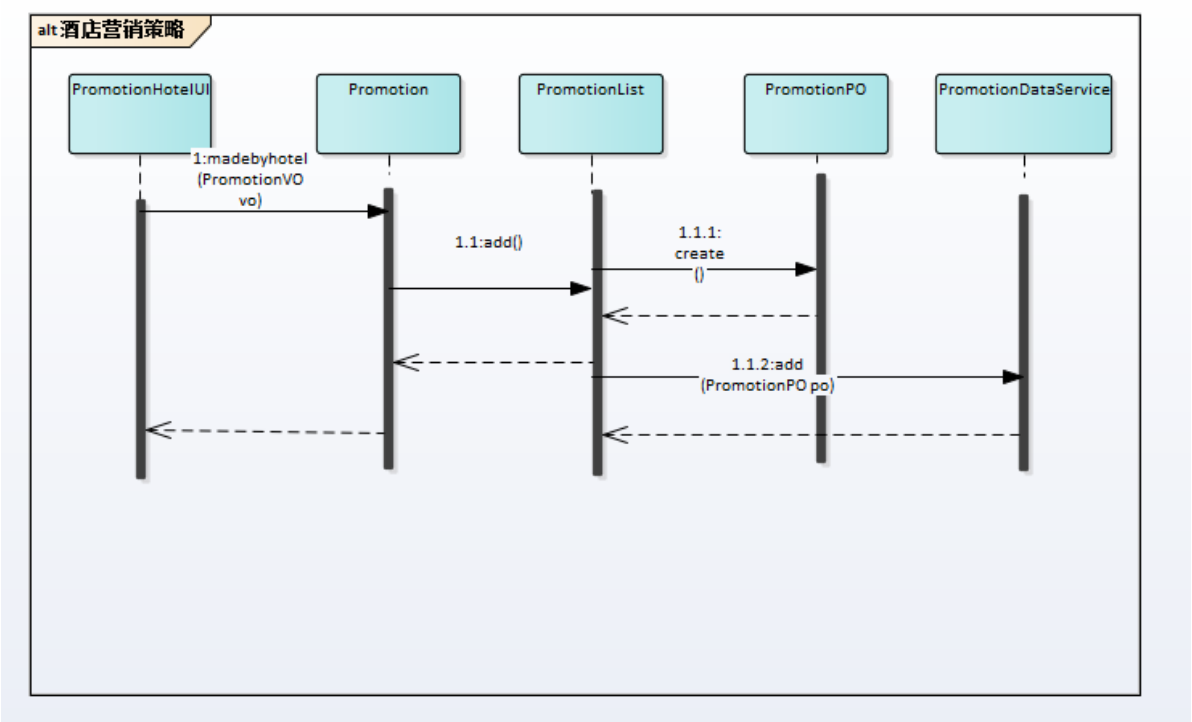

图 4-1-4-4 制定酒店营销策略的顺序图

图 4-1-4-5 表明了酒店预订系统中，在取消制定营销策略时，取消营销策略业务逻辑处理的相关对象之间的协作。

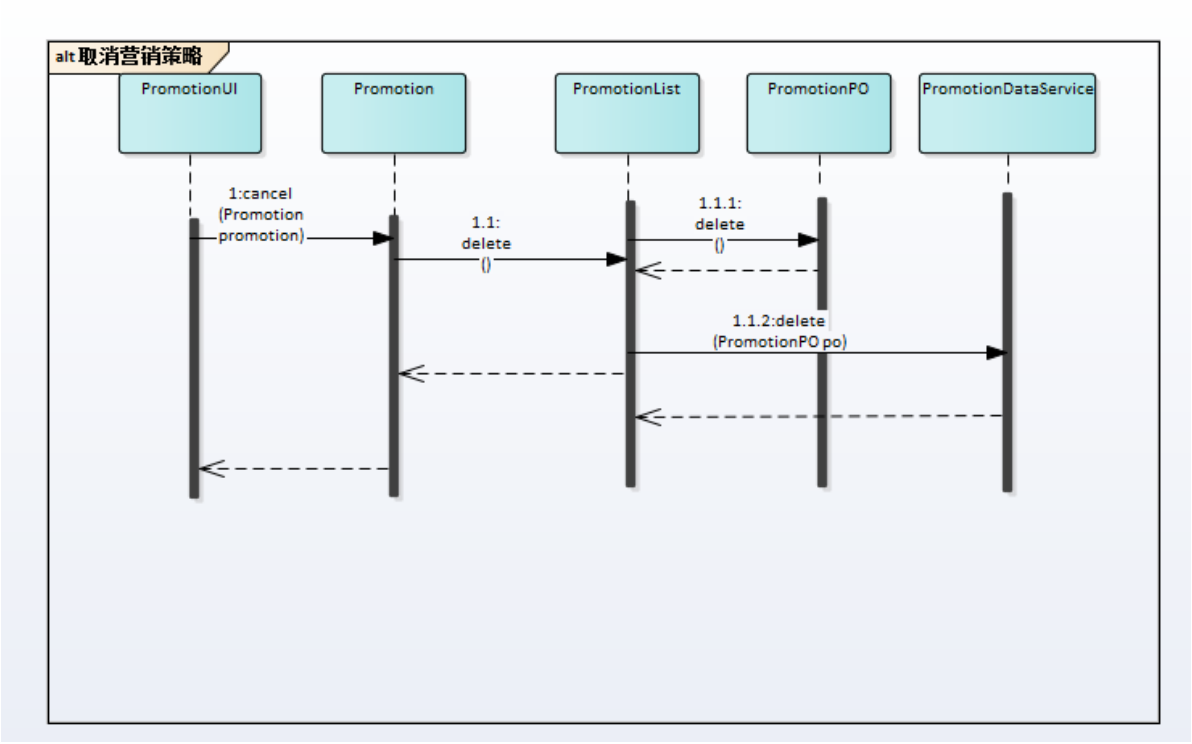

图 4-1-4-5 取消制定营销策略的顺序图

图 4-1-4-6 表明了酒店预订系统中，在制定网站营销策略时，制定网站营销策略业务逻辑处理的相关对象之间的协作。

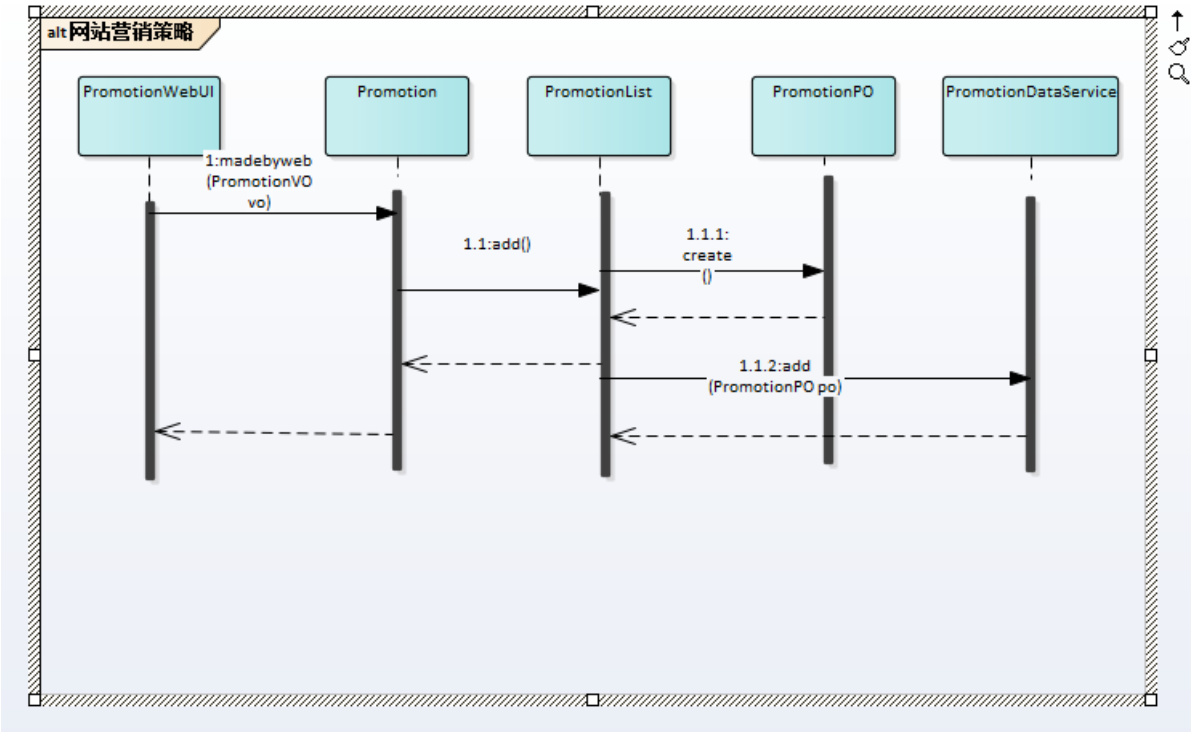

图 4-1-4-6 制定网站营销策略的顺序图

如图 4-1-4-7 所示的状态图描述了 Promotion 对象的生存期间的状态序列，引起转移的事件，以及因状态转移而伴随的动作。如果 addPromotion 方法被 UI 调用，

进入 Add 状态，之后通过 delete 进入 Delete 状态，最后通过 end 进入结束状态，如果处于开始状态时 delete 方法被调用，Promotion 进入 Delete 状态，最后通过 end 进入结束状态。

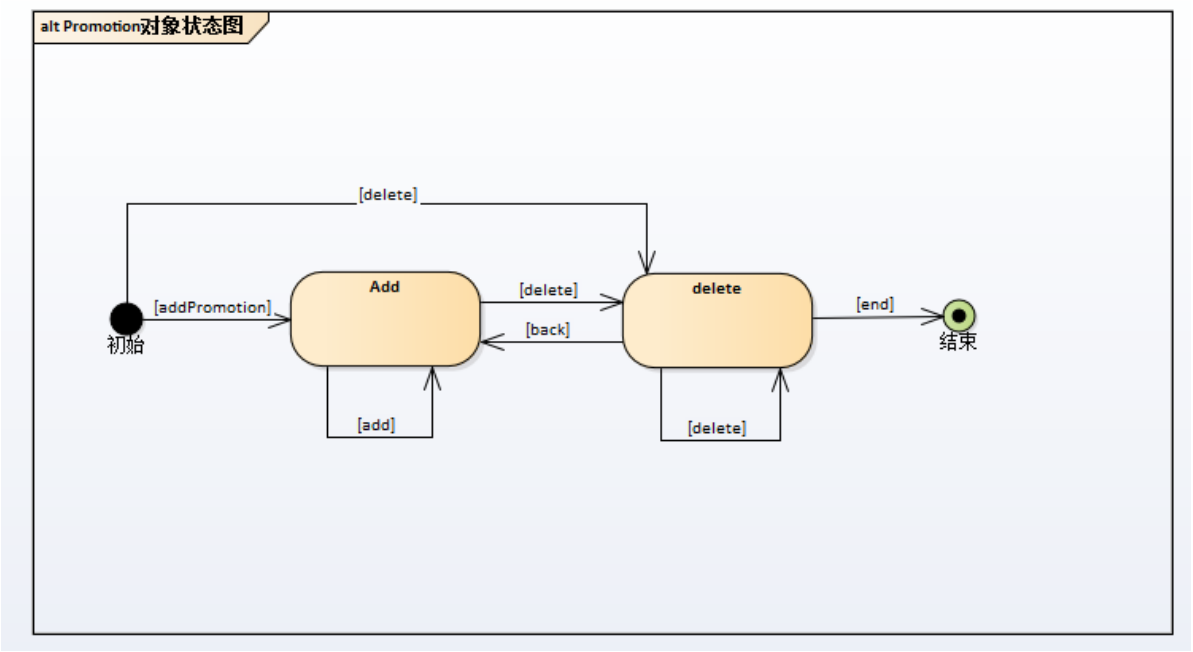

图 4-1-4-7 Promotion 对象状态图

4.1.5. userbl 模块

4.1.5.1. 模块概述

userbl 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。

userbl 模块的职责及接口参见软件系统结构设计文档。

4.1.5.2. 整体结构

根据体系结构的设计，酒店预订系统选择分层体系结构风格，将系统分为 3 层（展示层、业务逻辑层、数据层）,很好地示意了整个高层抽象。展示层包含 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。每一层之间为了增加灵活性，添加了接口，比如在展示层和业务逻辑层之间添加 blservice.userblservice.UserBLService 接口，在业务逻辑层和数据层之间添加 dataservice.userdataservice.UserDataService 接口。

UserPO 、 CreditRecordPO 、 OrderPO

是作为不同单据的持久化对象被添加到设计模型中去的。 PromotionInfo、 OrderInfo

都是根据依赖倒置原则，为了消除循环依赖而产生的接口。

userbl 模块的设计如图 4-1-5-1 所示。

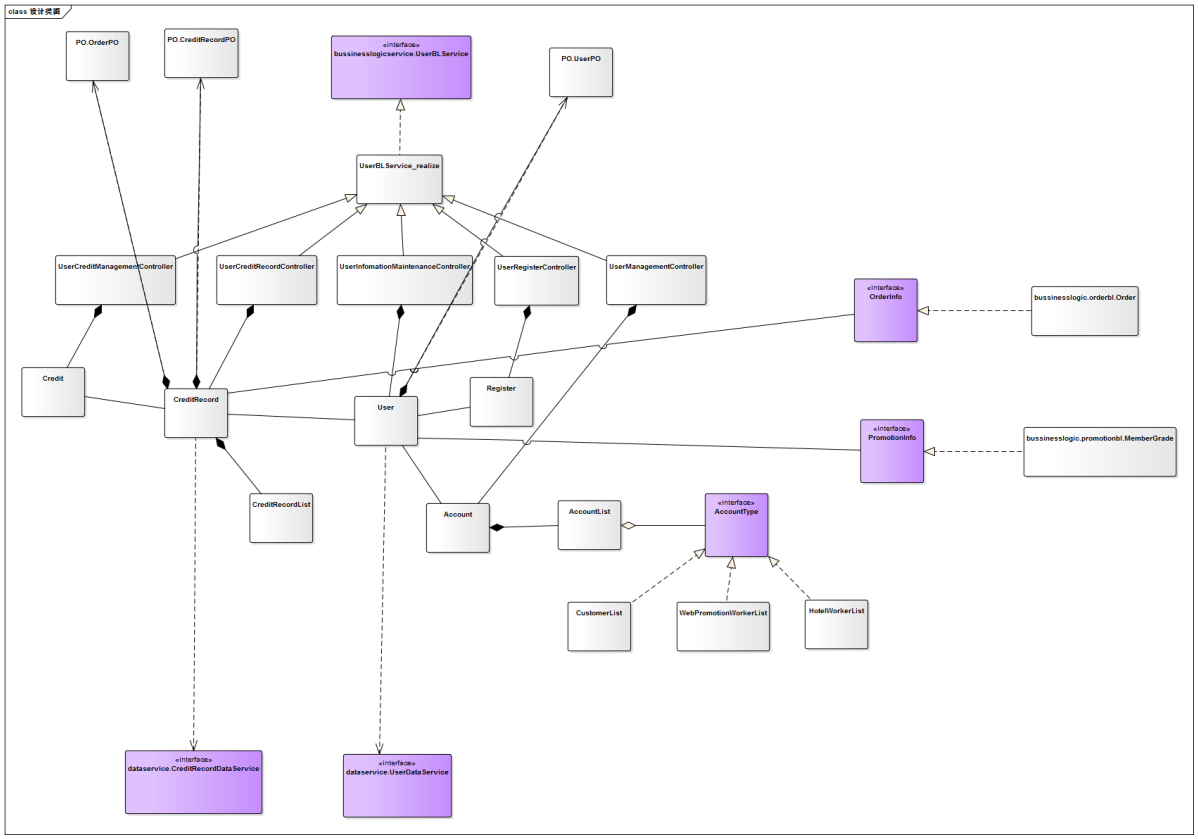

图 4-1-5-1 userbl 模块各个类的设计

userbl 模块各个类的职责如表 4-1-5-1 所示。

表 4-1-5-1 userbl 模块各个类的职责

类	职责

UserBIService_realize	负责实现与用户有关界面所需要的服务
UserCreditManagementController	负责实现对应于客户信用值管理界面所需要的服务
UserCreditRecordController	负责实现对应于信用记录界面所需要的服务
UserInformationMaintenanceController	负责实现对应于用户信息维护界面所需要的服务
UserRegisterController	负责实现对应于注册界面所需要的服务
UserManagementController	负责实现对应于账户管理所需要的服务
Credit	信用管理的领域模型对象，拥有用户账号，充入金额，当前信用值等数据，可以帮助完成信用管理界面所需要的服务
CreditRecord	信用记录的领域模型对象，拥有记录时间，动作（订单执 行，订单异常，订单撤销，充值），信用度变化， 当前信用值，可以帮助完成信用记录界面所需要的服务

Register	注册的领域模型对象，拥有注册用户的用户名、账户、密码，可以解决注册问题
User	系统用户的领域模型对象，拥有客户名称，客户账号，客户联系方式， 会员等级，会员种类，生日，企业名称，可以帮助完成用户信息维护所需要的服务
CreditRecordList	封装关于 CreditRecord 的数据集合的数据结构的秘密，可以帮助完成信用记录界面所需要的服务
Account	账户的领域模型对象
AccountList	封装关于 Account 的数据集合的数据结构的秘密，可以帮助完成账户管理所需要的服务
CustomerList	封装关于 CustomerAccount 的数据集合的数据结构的秘密，可以帮助完成账户管理所需要的服务

WebPromotionWorkerList	封装关于WebPromotionWorkerAccount的数据集合的数据结构的秘密，可以帮助完成账户管理所需要的服务
HotelWorkerList	封装关于 HotelWorkerAccount的数据集合的数据结构的秘密，可以帮助完成账户管理所需要的服务

4.1.5.3. 模块内部类的接口规范

UserBL 的接口规范如表 4-1-5-2 所示。

User 的接口规范

提供的服务（供接口）		
User.findByID	语法	**public** UserVO findByID(String userID)
	前置条件	无
	后置条件	返回属于此 ID 的用户信息

User.updateLevel	语法	**public void** updateLevel(UserVO vo)
	前置条件	存在此用户
	后置条件	更新此用户会员等级
User.logout	语法	**public void** logout(String id)
	前置条件	用户已登录
	后置条件	用户为登出状态
User.login	语法	**public void** login(String id)
	前置条件	用户未登录

	后置 条件	用户为登录状态
User.updateUserInfo	语法	**public void** updateUserInfo(UserVO vo)
	前置 条件	存在此用户
	后置 条件	更新用户信息
User.create	语法	**public void** create(UserVO vo)
	前置 条件	无
	后置 条件	创建一个新用户
需要的服务（需接口）		
RemoteHelper.*getInstance*().getDataFactoryService()		得到数据工厂服务
MemberGrade.getLevelSystem()		得到会员等级系统
DataFactoryService.getDataService(String type)		得到 User 数据服务

UserDataService.update(UserPO po)	更新单一持久化对象
UserDataService.insert(UserPO po)	插入单一持久化对象

Register 的接口规范

提供的服务（供接口）		
Register.add	语法	**public void** add(UserVO vo)
	前置条件	无
	后置条件	增加一个新用户
Register.getUser	语法	**public** UserVO getUser(String id)
	前置条件	存在此用户

	后置条件	返回此用户信息
需要的服务（需接口）		
User.create(UserVO vo)		创建一个新用户
User.finByID(String id)		按 ID 查找用户

CreditRecord 的接口规范

提供的服务（供接口）		
CreditRecord.showCreditRecord	语法	**public** HashMap<String,CreditRecordVO> showCreditRecord(String id)
	前置条件	存在此用户
	后置条件	返回此用户所有信用记录

CreditRecord.updateCreditRecord	语法	**public void** updateCreditRecord(String id,CreditRecordVO vo)
	前置条件	存在此用户
	后置条件	更新此用户信用记录
CreditRecord.add	语法	**public void** add(String userID,CreditRecordVO vo)
	前置	无

	条件	
	后置条件	增加此用户的信用记录
CreditRecord.getCreditRecord	语法	**public** CreditRecordVO getCreditRecord(String userID,String orderID)
	前置条件	存在此信用记录
	后置条件	返回属于此用户和记录编号的信用记录
需要的服务（需接口）		
RemoteHelper.*getInstance*().getDataFactoryService()		得到数据工厂服务

DataFactoryService.getDataService(String type)	得到 CreditRecord 数据服务
CreditRecordDataService.update(CreditRecordPO po)	更新单一持久化对象
CreditRecordDataService.insert(CreditRecordPO po)	插入单一持久化对象

Credit 的接口规范

提供的服务（供接口）		
Credit.showCredit	语法	**public long** showCredit(String id)
	前置条件	存在此用户
	后置条件	返回此用户信用值
Credit.updateCredit	语法	**public void** updateCredit(String

		userID,CreditRecordVO vo)
	前置条件	存在此用户
	后置条件	更新此用户信用值
需要的服务（需接口）		
CreditRecord. add(String userID,CreditRecordVO vo)		增加此用户的信用记录

Account 的接口规范

提供的服务（供接口）		
Account.getUser	语法	**public** UserVO getUser(String account)
	前置条件	此账号存在
	后置条件	返回此账号信息

Account.update	语法	**public void** update(UserVO vo)
	前置条件	此账号存在
	后置条件	更新此账号信息
Account.delete	语法	**public void** delete(UserVO vo)
	前置条件	此账号存在
	后置条件	删除此账号
Account.add	语法	**public void** add(UserVO vo)

	前置条件	无
	后置条件	增加一个账户
需要的服务（需接口）		
AccountList.getAccountList(int type)		得到对应类型的账号列表
RemoteHelper.*getInstance*().getDataFactoryService()		得到数据工厂服务
DataFactoryService.getDataService(String type)		得到 User 数据服务
UserDataService.update(UserPO po)		更新单一持久化对象
UserDataService.insert(UserPO po)		插入单一持久化对象
UserDataService.delete(UserPO po)		删除单一持久化对象

表 4-1-5-2 Userbl 的接口规范

4.1.5.4. 业务逻辑层的动态模型

图 4-1-5-2 表明了酒店预订系统中，在查找客户时输入一个 ID 后，查找客户业务逻辑处理的相关对象之间的协作。

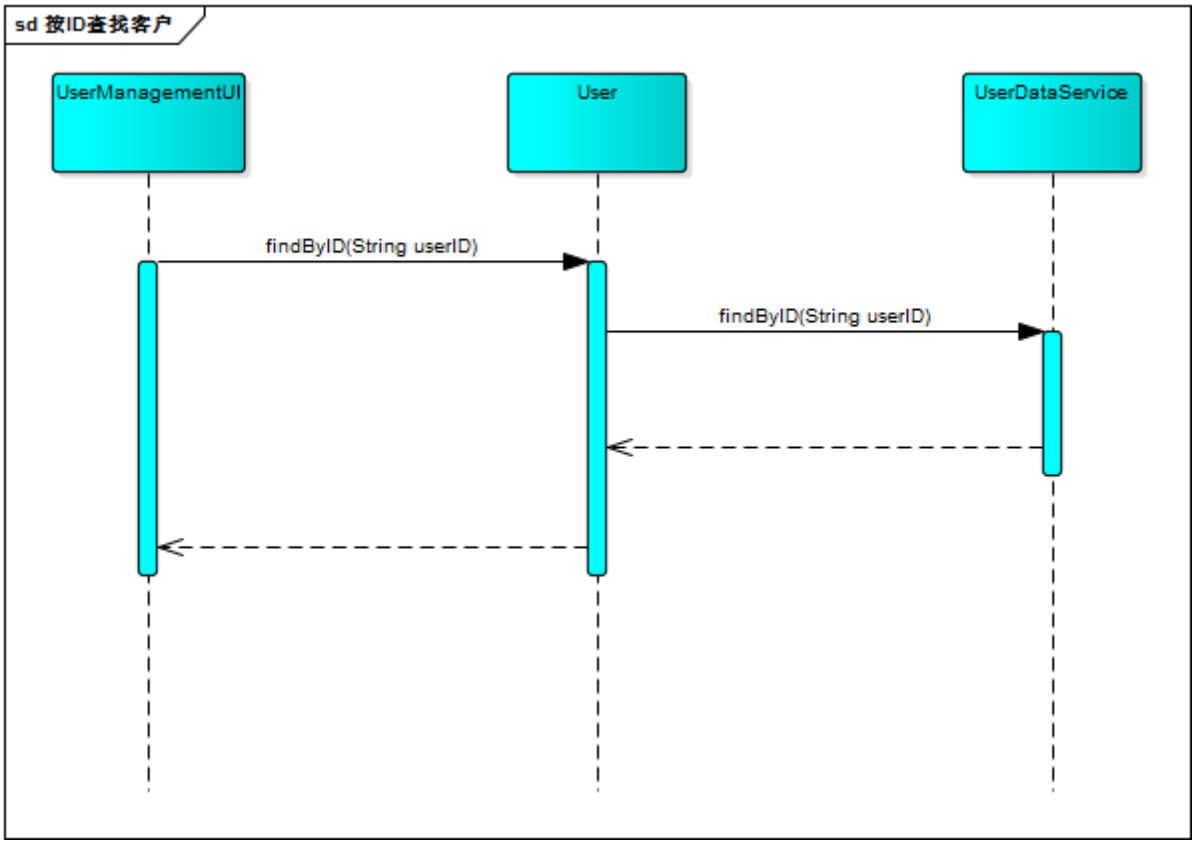

图 4-1-5-2 按 ID 查找客户的顺序图

图 4-1-5-3 表明了酒店预订系统中，在客户登出时，登出业务逻辑处理的相关对象之间的协作。

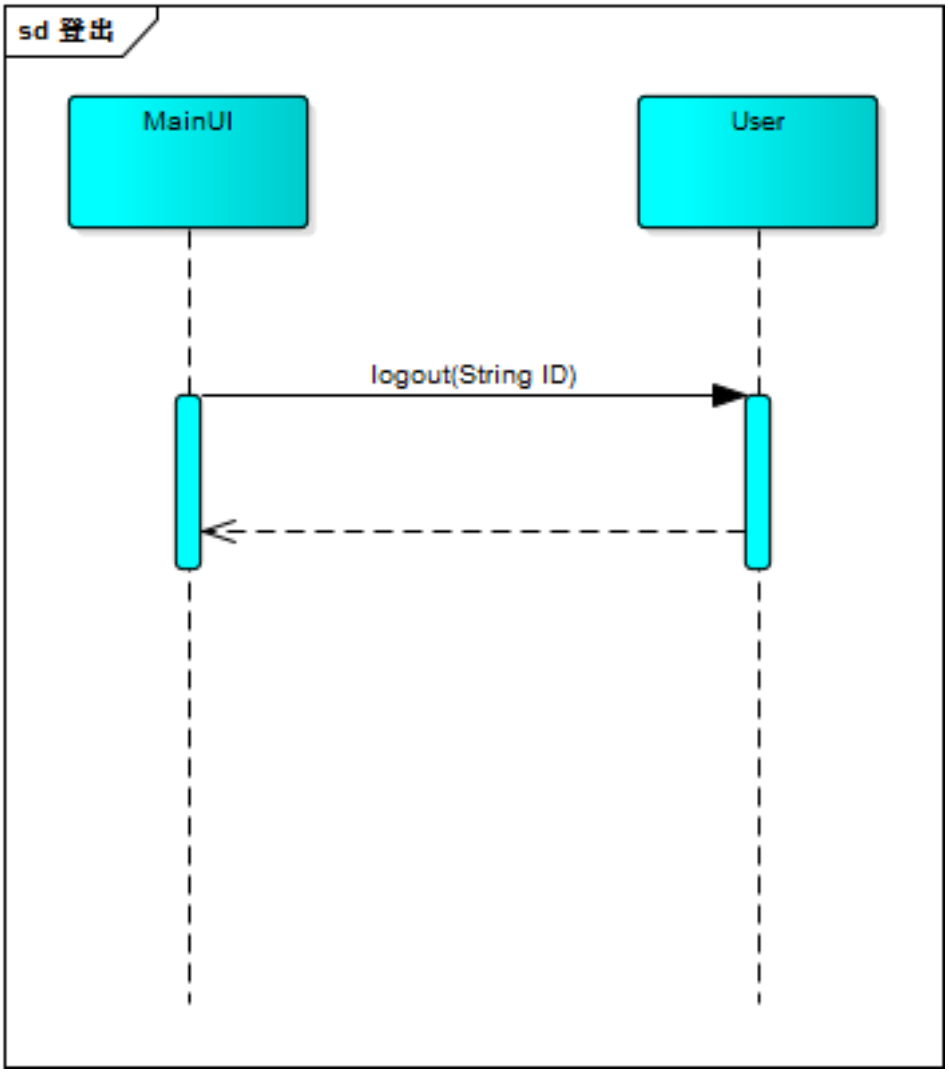

图 4-1-5-3 登出的顺序图

图 4-1-5-4 表明了酒店预订系统中，登录时，登录业务逻辑处理的相关对象之间的协作。

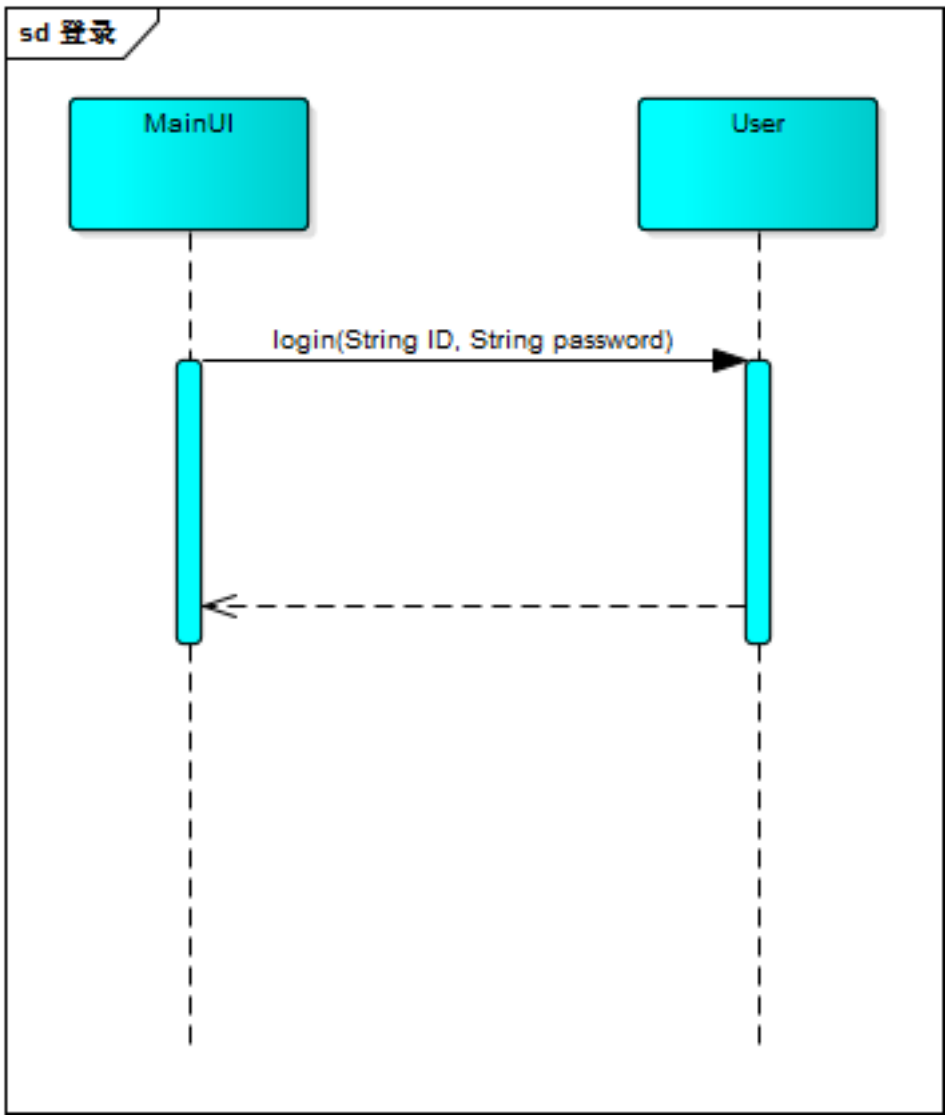

图 4-1-6-4 登录的顺序图

图 4-1-6-5 表明了酒店预订系统中，更新客户会员等级时，更新客户会员等级业务逻辑处理的相关对象之间的协作。

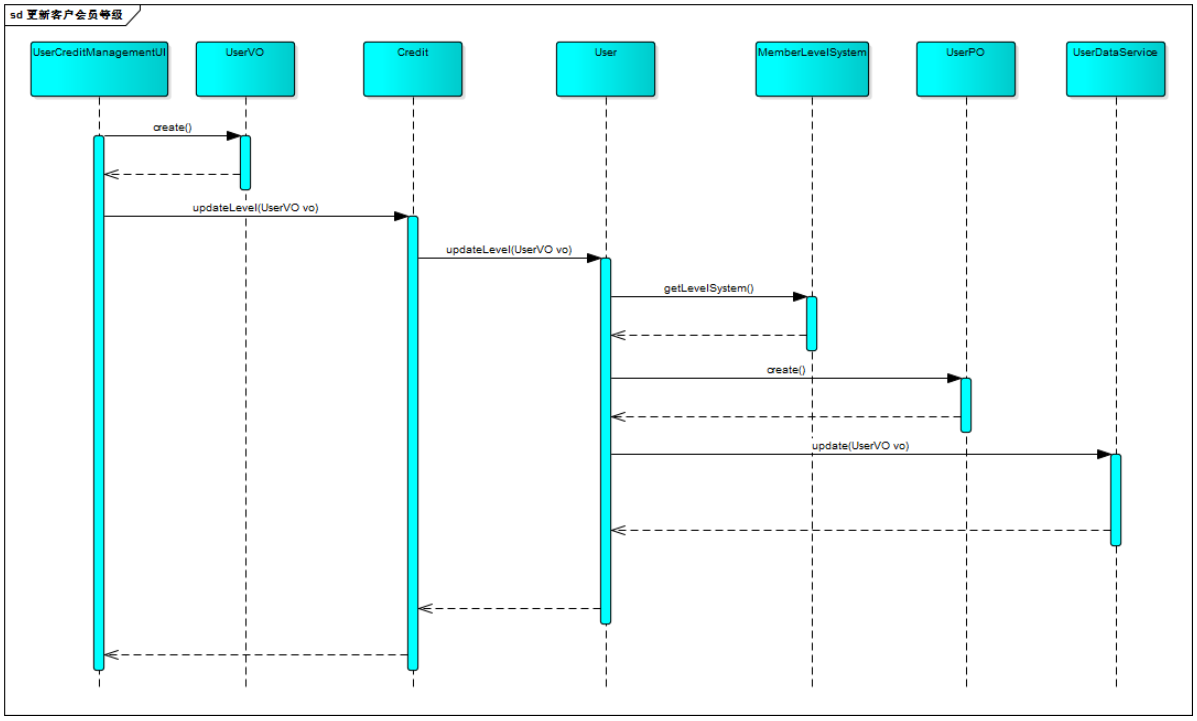

图 4-1-5-5 更新客户会员等级的顺序图

图 4-1-5-6 表明了酒店预订系统中，更新客户信息时，更新客户信息业务逻辑处理的相关对象之间的协作。

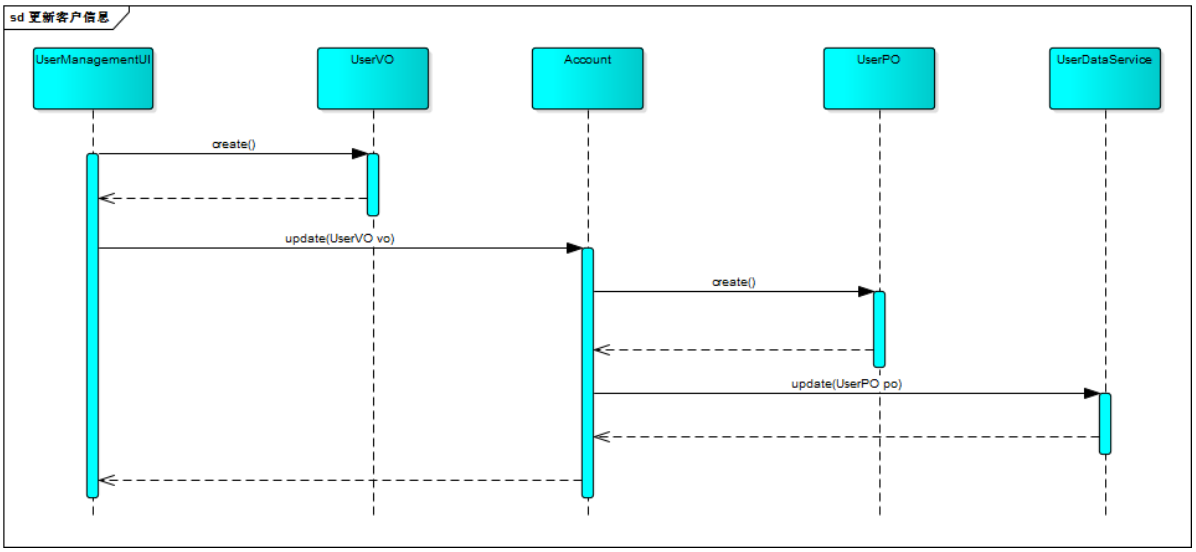

图 4-1-5-6 更新客户信息的顺序图

图 4-1-5-7 表明了酒店预订系统中，在更新信用记录时，更新信用业务逻辑处理的相关对象之间的协作。

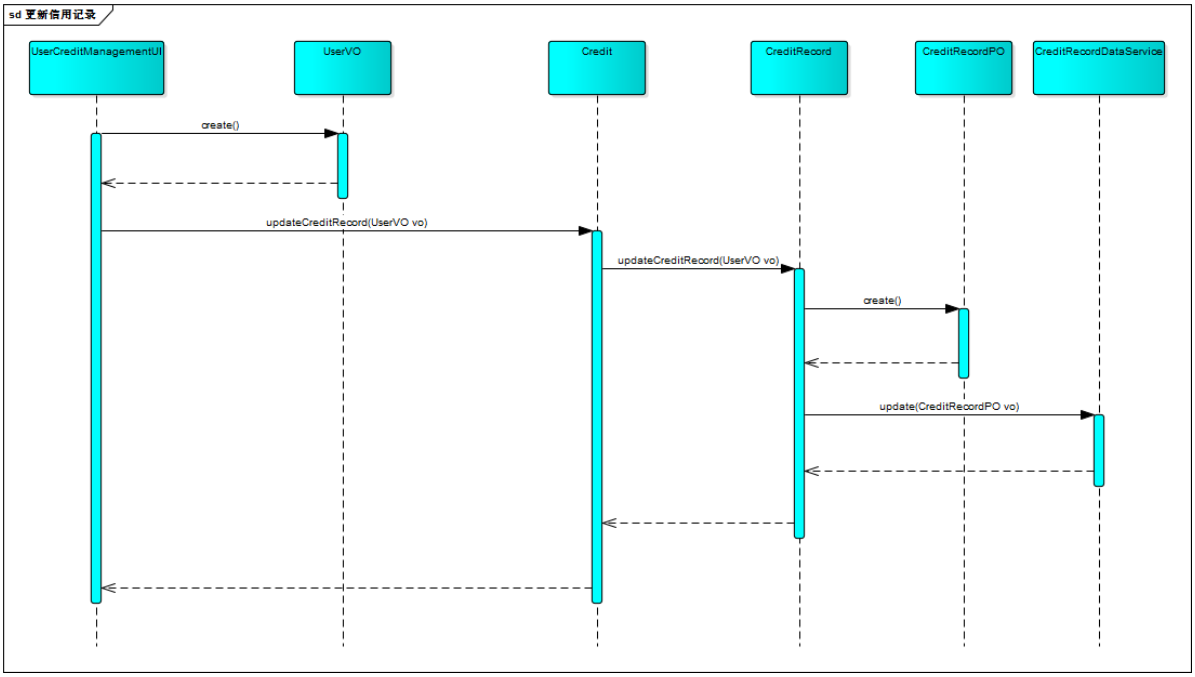

图 4-1-5-7 更新信用记录的顺序图

图 4-1-5- 8 表明了酒店预订系统中，在更新信用值时，更新信用业务逻辑处理的相关对象之间的协作。

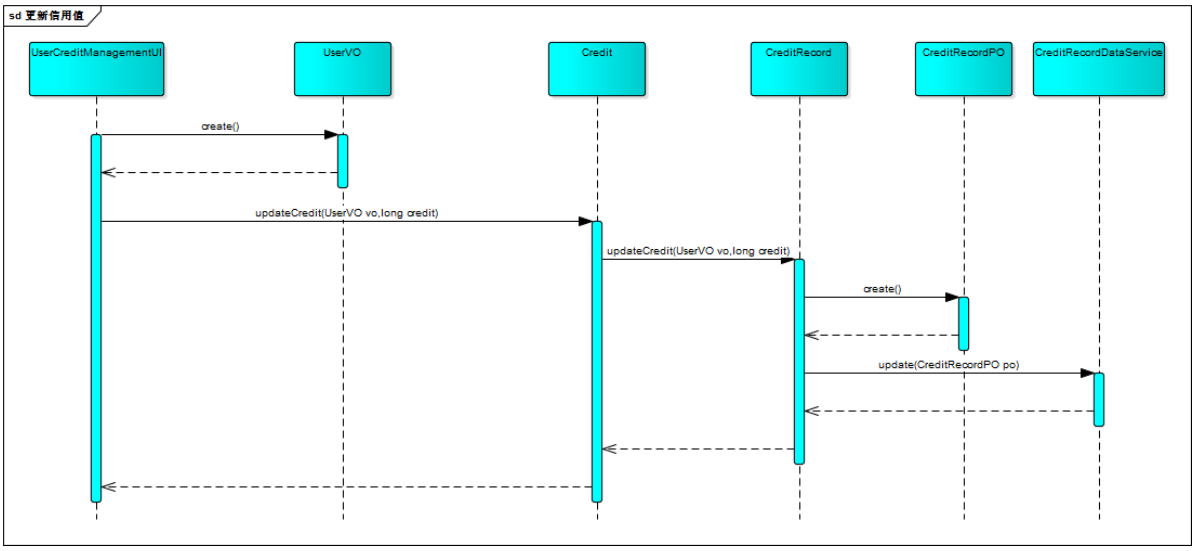

图 4-1-5- 8 更新信用值的顺序图

图 4-1-5- 9 表明了酒店预订系统中，在删除客户时，删除客户业务逻辑处理的相关对象之间的协作。

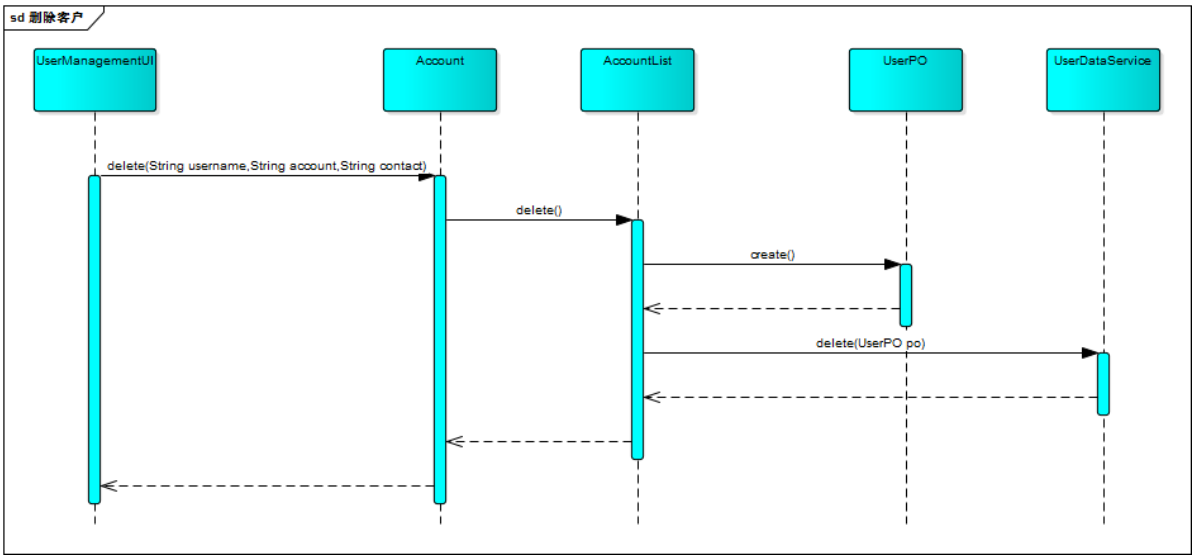

图 4-1-5-9 删除客户的顺序图

图 4-1-5- 10 表明了酒店预订系统中，在添加客户时，添加客户业务逻辑处理的相关对象之间的协作。

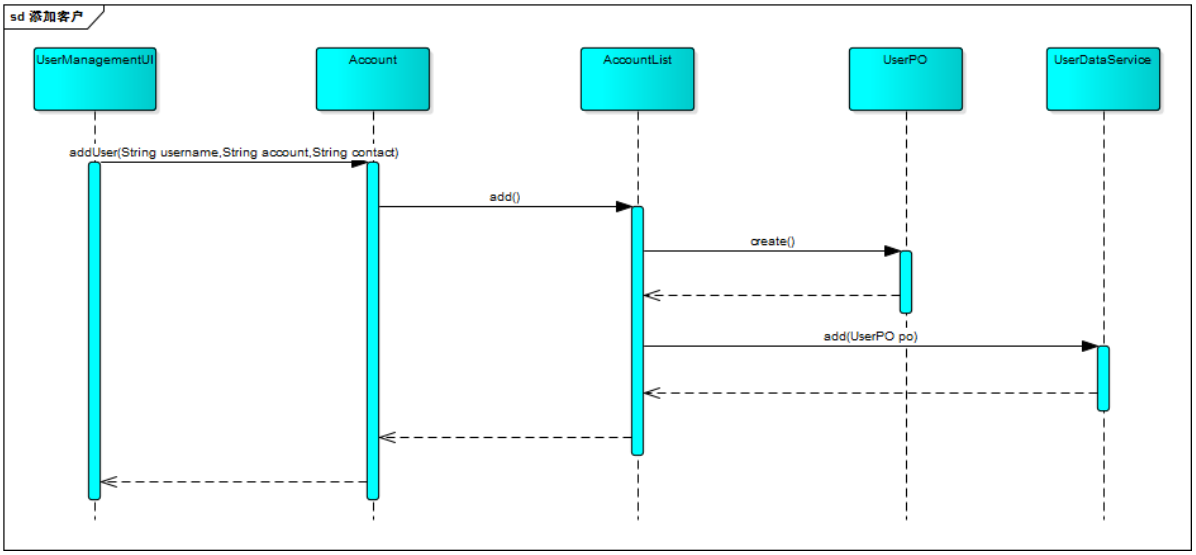

图 4-1-5-10 添加客户的顺序图

图 4-1-5- 11 表明了酒店预订系统中，在维护客户基本信息时，维护客户信息业务逻辑处理的相关对象之间的协作。

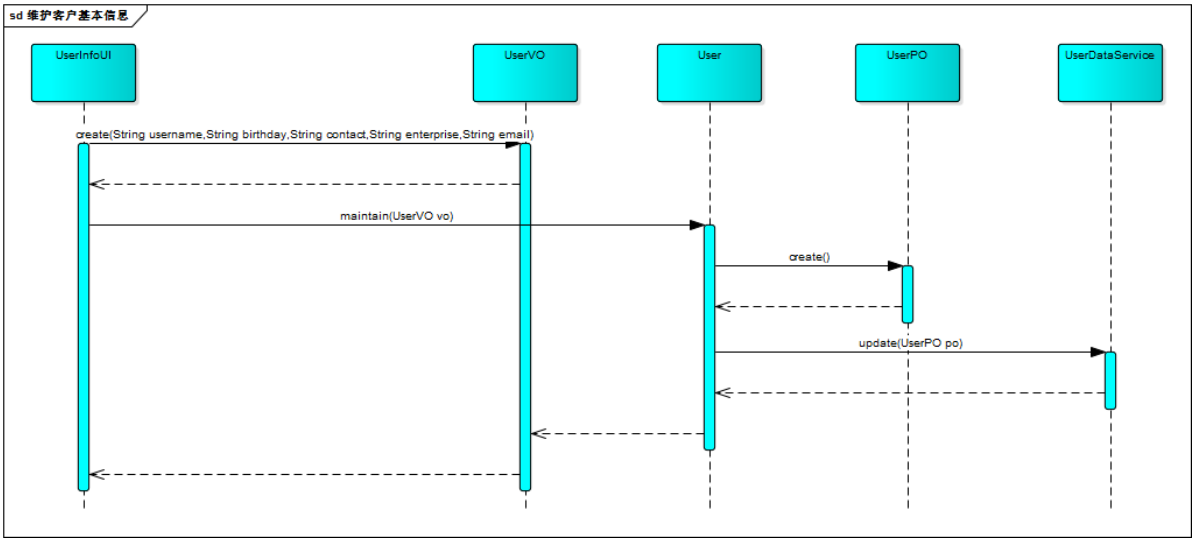

图 4-1-5-11 维护客户基本信息的顺序图

图 4-1-5- 12 表明了酒店预订系统中，在显示信用记录时，显示信用记录业务逻辑处理的相关对象之间的协作。

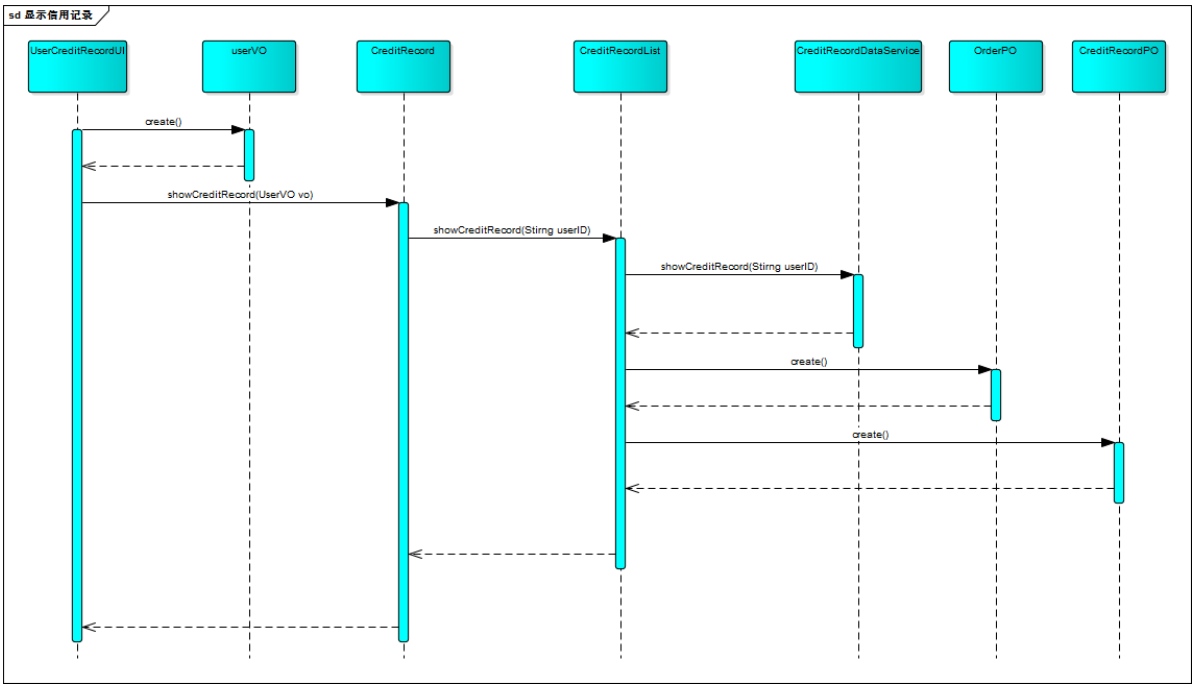

图 4-1-5-12 显示信用记录的顺序图

图 4-1-5- 13 表明了酒店预订系统中，在显示信用值时，显示信用值业务逻辑处理的相关对象之间的协作。

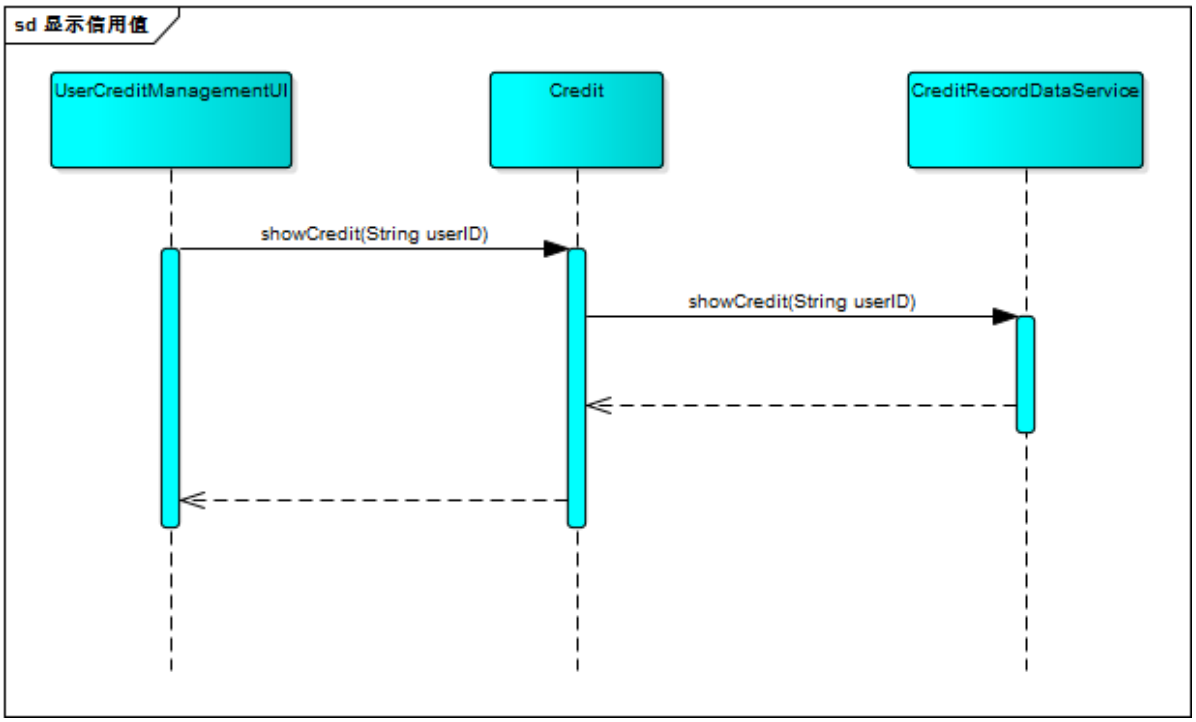

图 4-1-5-13 显示信用值的顺序图

图 4-1-5- 14 表明了酒店预订系统中，在客户注册时，注册业务逻辑处理的相关对象之间的协作。

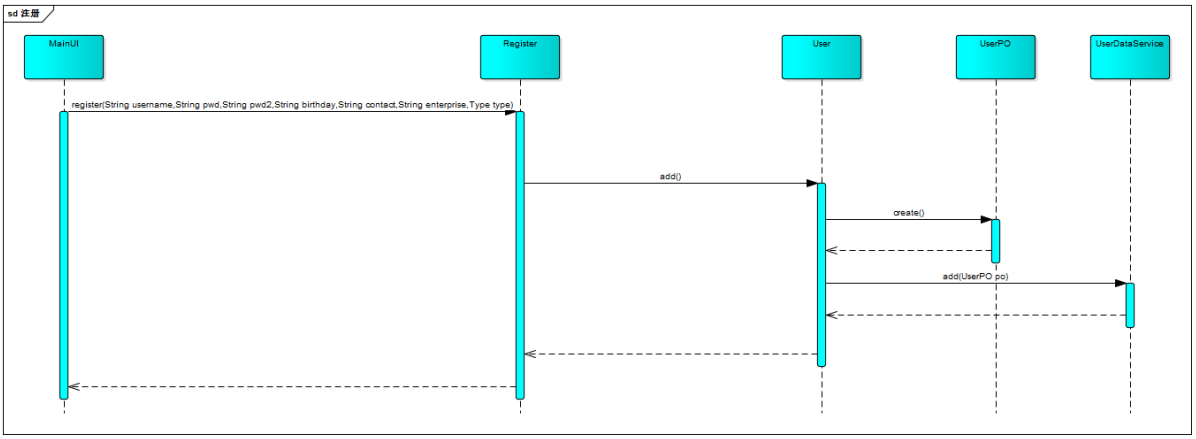

图 4-1-5-14 注册的顺序图

如图 4-1-5-15 所示的状态图描述了 Account 对象的生存期间执行查询单据操作的状态序列，引起转移的事件，以及因状态转移而伴随的动作。如果 addUser 方法被 UI 调用，Account 进入 Add 状态，然后如果 update 方法被调用，则进入 Update 状态，然后如果 delete 方法被调用，则进入 Delete 状态，最后通过 end 进入结束状态，处于开始状态时，如果调用 update 方法，则直接进入 Update 状态，如果调用 delete，则直接进入 Delete 状态。

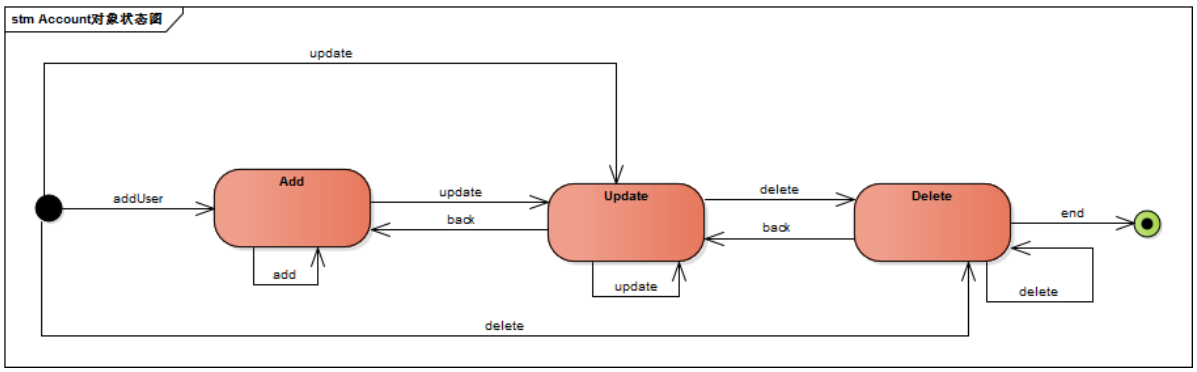

图 4-1-5-15 Account 对象执行查询操作的状态图

如图 4-1-5-16 所示的状态图描述了 CreditRecord 对象的生存期间执行增加单据操作的状态序列，引起转移的事件，以及因状态转移而伴随的动作。如果 showCreditRecord 方法被 UI 调用，CreditRecord 进入 Presentation 状态，然后如果调用 updateCreditRecord 方法，则进入 Updaet 状态， 最后通过 end 进入结束状态。如果处于开始状态，调用 updateRecord 方法，则直接进入 Update 状态。

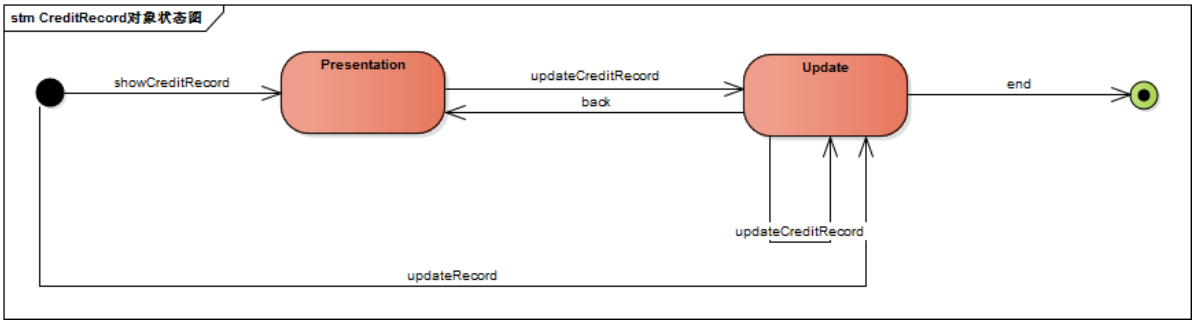

图 4-1-5-16 CreditRecord 对象执行增加操作的状态图

如图 4-1-5-17 所示的状态图描述了 Credit 对象的生存期间执行删除单据信息操作的状态序列，引起转移的事件，以及因状态转移而伴随的动作。如果 show 方法被 UI 调用，Credit 进入 Presentation 状态，然后 updateCredit 方法被 UI 调用，Credit 进入 Update 状态，然后可以调用 updateLevel 方法进入

Level 状态，也可以调用 updateCreditRecord 方法进入 CreditRecord 状态，也可以调用 end 方法进入结束状态，如果处于开始状态，可以调用 updateCredit 方法直接进入 Update 状态。

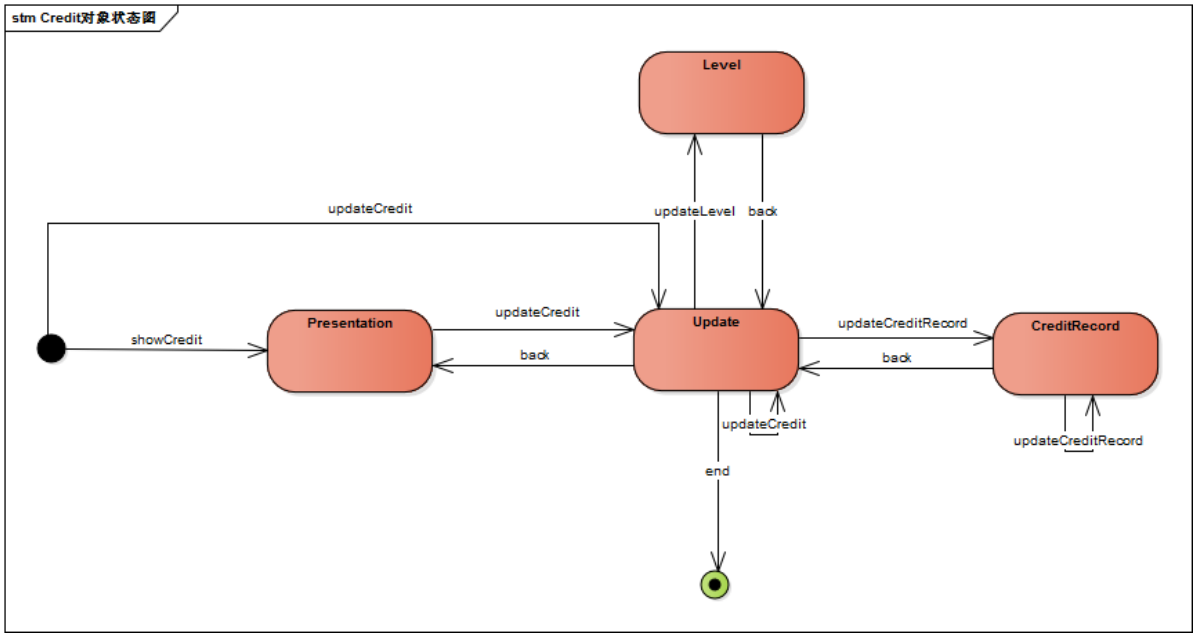

图 4-1-5-17 Credit 对象执行删除操作的状态图

如图 4-1-5-18 所示的状态图描述了 User 对象的生存期间执行更新单据操作的状态序列，引起转移的事件，以及因状态转移而伴随的动作。如果 register 方法或者 login 方法被 UI 调用，User 进入 Member 状态，如果 maintainPersonalInfo 方法被调用，则进入 Maintain 状态，如果 updataLevel 方法被调用，则进入 Level 状态，如果 logout 方法被调用，则进入结束状态。

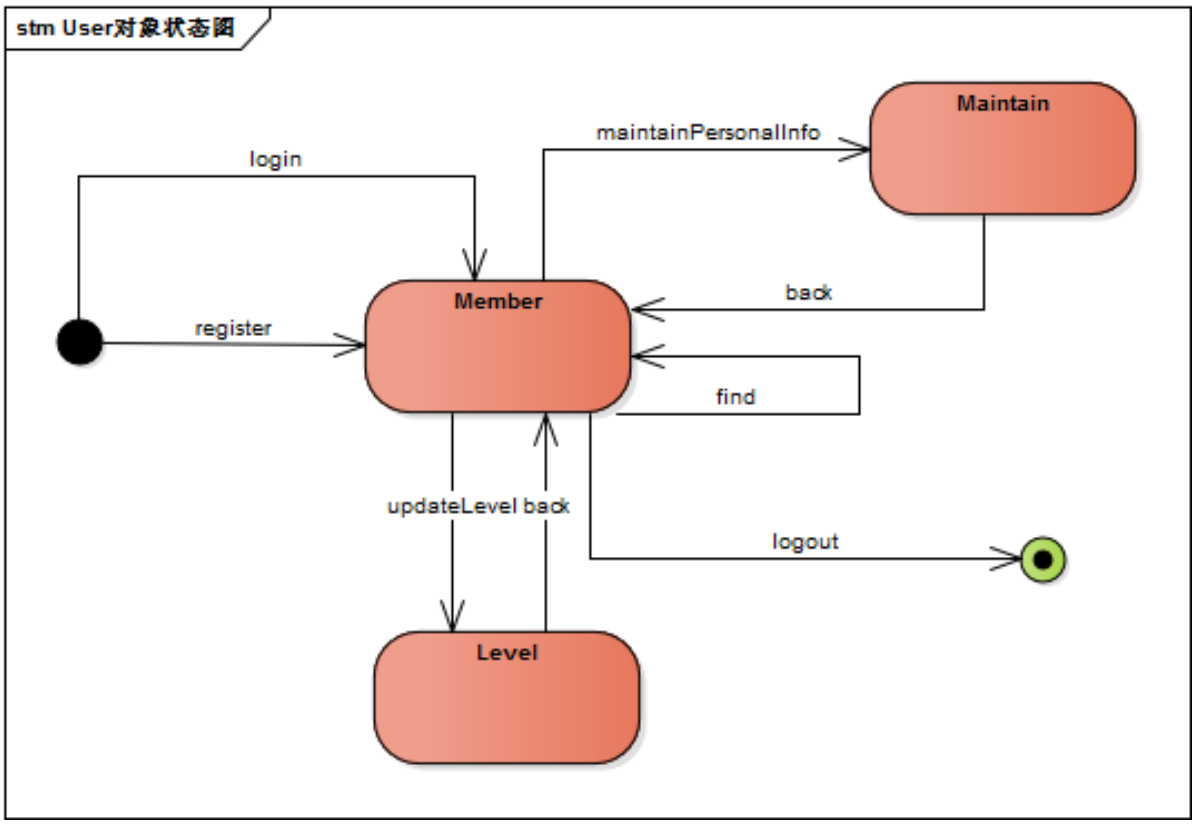

图 4-1-5-18 User 对象执行更新操作的状态图

5. 依赖视角

图 5-1 和图 5-2 是客户端和服务器端各自的包之间的依赖关系。

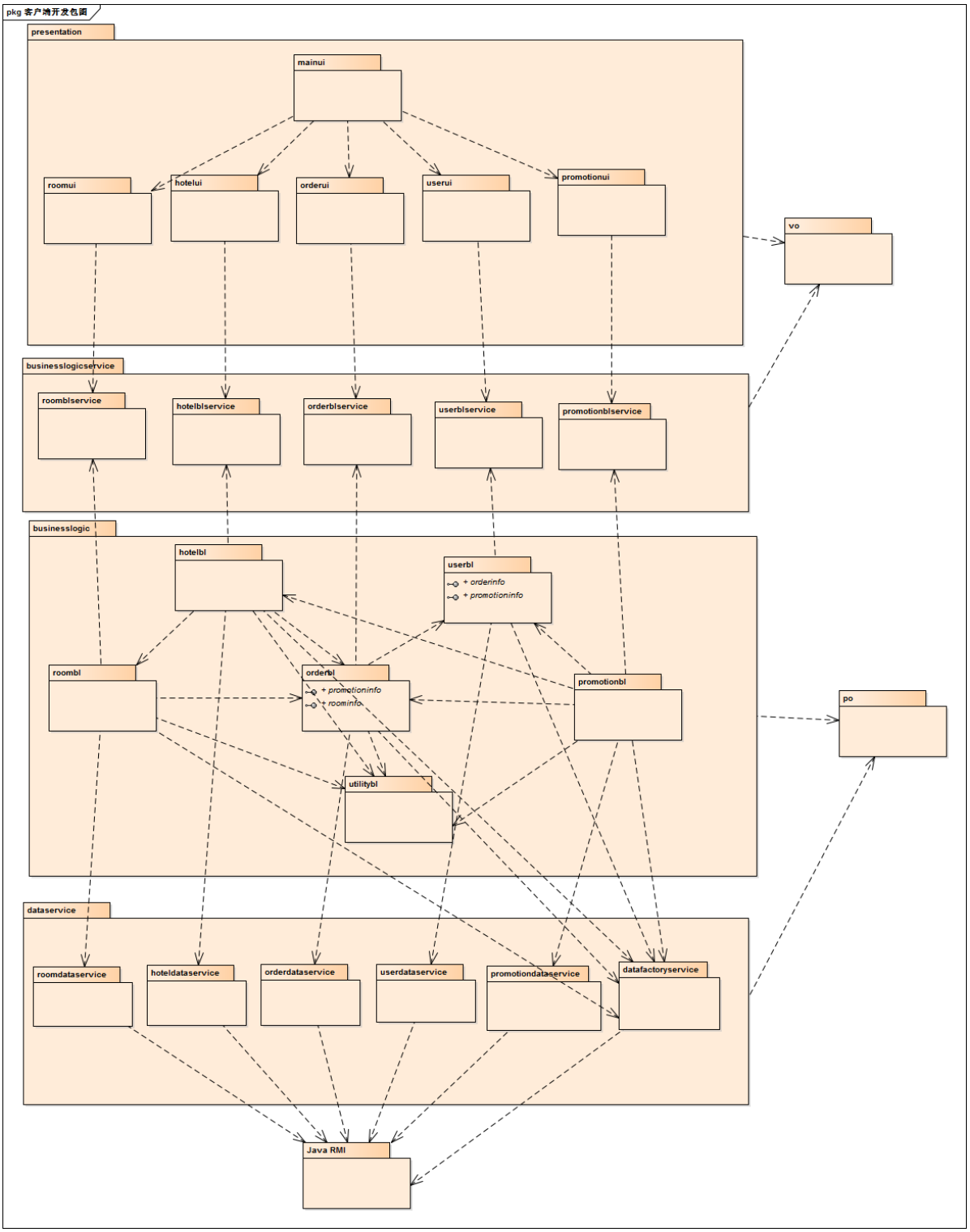
pkg 客户端开发包图
presentation
mainui
roomui
hotelui
orderui
userui
promotionui
vo
businesslogicservice
roomblservice
hotelblservice
orderblservice
userblservice
promotionblservice
businesslogic
hotelbl
userbl
+ orderinfo
+ promotioninfo
roombl
orderbl
+ promotioninfo
+ roominfo
promotionbl
po
utilitybl
dataservice
roomdataservice
hoteldataservice
orderdataservice
userdataservice
promotiondataservice
datafactoryservice
Java RMI

图 5-1 客户端包图

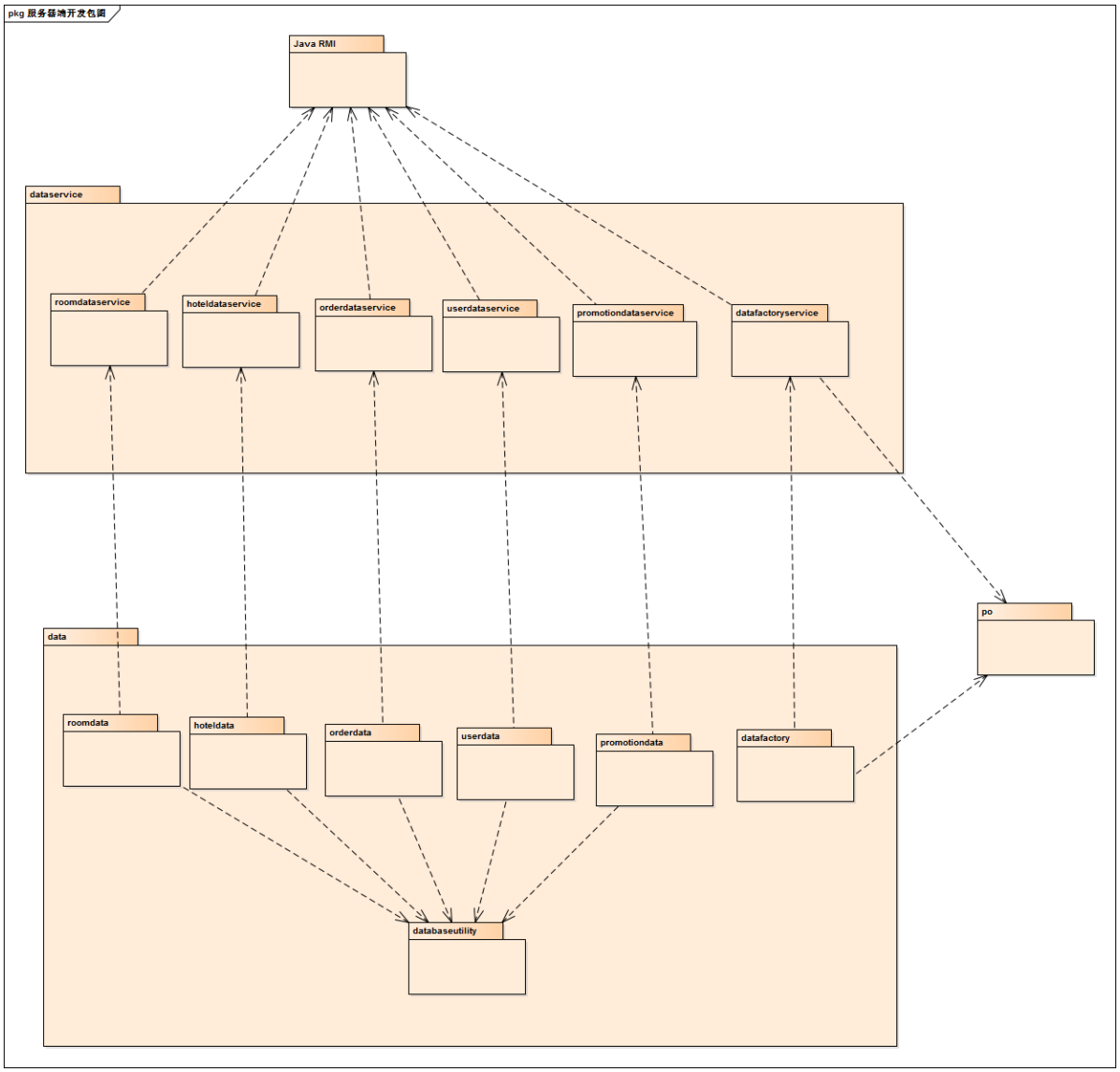

图 5-2 酒店预订系统服务器端开发包图