

1.Introduction

1.1 系统目标

旨在设计一个宾馆线上预订网站。其中要涉及四个方面。

(1)普通旅客可以通过网站查询, 预订酒店。也可以操作自己的订单。

(2)网站不但面向普通的客户, 还要照顾到旅店的管理人, 方便旅店管理人操作自己接收到的订单。

(3)网站管理员要处理投诉等客户的请求, 要能对网站的订单进行直接的操作。

(4)向旅客推荐合适的酒店。

1.2 设计目标

与分析不同, 系统设计是开发者的领域。客户和最终客户将退至幕后。因此, 我们有了以下几个设计目标:

(1)低操作成本。为了减少开销, 运行系统的成本应该最小化

(2)高可用性。随着旅客数目的增加, 系统的负担也会加大。这个设计目标在需求分析中非功能性需求中“可用性”中隐性的提出来了。

(3)由旅客和同时进行的活动数目得出可测量性。旅店预订系统的响应时间不能随着读者 User 数目的增加大幅度的变长。当需要的时候, 管理员 Adminuser 应该能够选择增加硬件结点来扩充系统的容量。这个设计目标也在需求分析中非功能性需求中“可测量性”中提出来了。

(4)为开放资源开发进行编档。这个设计目标来自开发者和管理的, 与客户相反。

1.3 参考文献

《面向对象软件工程》

2.Proposed System

2.1Overview

因为这是一个全新的软件体系, 因此需要从头至尾全面分析功能需求,非功能需求, 系统模式和限制条件。

我们应当把系统区分为 3 部分, 旅客部分, 旅馆部分, 和旅店管理员部分。

2.2Functional Requirements

客户需求

一. 注册成为 Booking 用户

- 客户注册时提供姓名, 年龄, 手机号, 常用邮箱。(手机号和常用邮箱均需验证)
- 客户需绑定信用卡, 需给自己起一个昵称(可选, 默认为“游客”)
- 客户可以上传照片作为头像
- 客户需填写常用信息(真实姓名, 联系方式, 身份证号)作为预约酒店的必要信息
- 客户注册时填写的密码必须是 8 位以上 30 位以下, 只能包含大小写字母和数字, 并且大小写字母与数字必须均存在。

二. 登陆并管理个人中心

1. 登陆个人中心

- 输入账号密码
- 输入随机产生的验证码
- 如果忘记密码，点击『忘记密码』链接
- 点击『登陆』按钮

2. 编辑个人资料

- 上传『个人照片』填写昵称，昵称可以随时更改
- 选择『年龄段』

18-24 岁

25-34 岁

35-44 岁

45-54 岁

55-64 岁

65 岁以上

- 选择『国家/地区』

此处包含全世界 200 多个国家和地区

- 选择『称谓』

先生

女士

- 填写『名』
- 填写『姓』
- 选择『性别』

男性

女性

保密

- 添加『电邮地址』，最多可以保存 3 个邮箱地址
- 填写『地址』『公司地址』，地址都有更改按钮
- 『修改密码』按钮，点击按钮，通过电子邮件提供重设密码的链接
- 此处提交的个人信息将和客户的评语一同显示，所有改动和更新将应用于所有点评。

3. 设置信用卡信息

- 填写信用卡号码，持卡人姓名，失效日期
- 有『添加卡片』按钮
- 最多绑定三个信用卡

4. 设置旅行偏好

- 选择『是否吸烟』

禁烟

吸烟

- 选择『首选星级』

任何星级

五星级

四星级及以上

三星级及以上

二星级及以上

- 选择『货币』

含世界各国货币

- 添加『首选设施』(可多选)

适合残疾人士的设施

餐厅

上网服务

禁烟客房

健身中心

家庭间

准许携带宠物

停车场

Spa 及健康中心

无线网络

室内游泳池

机场班车

免费无线网络连接

酒吧

行李寄存

24 小时前台

- 创建『旅行团体』

输入团体名称

添加成人信息 (1-10 个)

添加儿童信息 (0-10 个)

5. 我的收藏

- 储存您和同行客人信息
- 储存喜欢的酒店

三. 搜索并预订酒店

- 用户可以根据以下信息来查询

入住时间 (详细到小时)

所属公司

酒店名字

房间价格

酒店星级

酒店位置

优势设施

- 在查询中加入模糊搜索的功能, 让用户在打错或少打一两个字的同时仍能查到相关酒店。
- 用户所能查询到的酒店信息包括各个房间的价格, 具体位置, 酒店联系方式, 其他用户对该酒店的评价, 定前必读, 酒店服务种类, 酒店的照片 (1 到 5 张)。
- 定前必读包括

酒店支持的入住时间 (精确到小时)

退房时间 (精确到小时),

预定和取消预定政策,

是否接受儿童入住,

是否提供加床服务,

是否允许携带宠物入住,

酒店接受的银行卡类型。

是否有浴室，

浴室的配件中是否有浴袍，浴巾，吹风机，

卧室是否有拖鞋，闹钟，

厨房是否有咖啡机，电烧水壶，

酒店是否提供早中晚餐，

早中晚餐的供应时间（精确到小时）。

- 用户在预订酒店的同时可以对酒店留言提出一些相关要求（如加床等），供酒店工作人员查看。
- 用户的支付方式包括

线上支付(需提供已绑定的有效可用的信用卡)，

线下支付(待用户到酒店后自行与酒店工作人员完成付款工作)

- 两种预定方式均需提供完整的个人信息（真实姓名，联系方式，身份证号）与入住时间，退房时间。
- 用户在预定酒店之后可以查看自己的订单状态（查看是否预订成功），在允许取消订单的酒店预定的可以取消预定，可以打印订单信息。

四. 点评酒店与旅行推荐

1. 点评

- 为保证评分及评语内容的时效性和相关性，所有评语总数超过 30 条的住宿均只显示最近 24 个月的住客评语。
- 客人只有在 Booking.com 上预订住宿并成功入住后，才能填写住客评语。

2. 推荐

- 旅友的目的地推荐

五. 其他

- 搜索折扣优惠
- 查看热门目的地
- 探索热门目的地，购物、美食、历史。。。
- 查看短租公寓及别墅
- 查看 Booking.com 商旅服务

酒店管理员需求

一. 开始注册

1. 填写联系信息

- 填写希望与 Booking 网站合作的酒店数量
- 选择住宿类型
- 填写住宿名称
- 填写联系人姓名
- 填写联系电子邮箱
- 点击『开始注册』按钮
- 如果上次注册未完成，可以选择继续注册

2. 填写基本信息

- 选择住宿类型，填写公寓总数
- 是否能提供网络

- 是否提供停车场
- 是否为客人提供早餐
- 是独立拥有公寓，还是隶属于某个住宿管理公司/集团
- 是否接受信用卡/借记卡（如果不接受，网站将告知客人仅接受现金）
- 选择国家或地区，填写城市和邮政编码
- 填写详细公寓地址
- 填写联系电话（以便 Booking.com 和酒店商议注册事宜）
- 点击『下一步』

3. 填写公寓详情

- 选择公寓的描述

一卧室公寓

一室公寓

三卧室公寓

两卧室公寓

享有花园景致的公寓

公寓

公寓-分层

公寓-地面楼层

公寓-带共用浴室

公寓-带桑拿

公寓-带淋浴

公寓-带阳台

公寓-带露台

公寓套房

复式公寓

山景公寓

标准公寓

游泳池公寓

海景一室公寓

海景公寓

湖景公寓

豪华公寓

阁楼

顶楼公寓

高级公寓

高级公寓-带桑拿

- 填写可入住的人数
- 填写 1 人入住的价格（CNY/晚）
- 点击『保存并继续』

4. 上传酒店照片

- 至少上传 1 张照片
- 点击『下一步』

5. 相关设置

- 注明酒店接受的付款方式（信用卡/借记卡），还是仅接受现金

- 是否有资质接受外宾
- 设置预付款及预订取消政策

客人在入住如前多少天可以免费取消？

若不可免费取消客人需支付多少金额？

是否要求客人在预订酒店时支付预付款？

- 设置税费

有无增值税

有无城市税

- 设置发票信息

选择对账单接收人

此接收人的地址和酒店地址是否一致？

- 是否通过渠道管理商或者直连供应商管理房价和房量？
- 点击『下一步』

6. 住宿合约

- 查看 Booking.com 与酒店的合约，同意就勾选 2 个选项

我保证此住宿为一家合法经营的住宿场所，具备所有必要的营业执照和许可，并可在首次接到要求时出示这些文件。Booking 保留验证及核查您在此次注册中提供的各类信息的权利。

我方已阅读、接受并同意条款与协议。

- 点击『下一步』

此时注册完成。可以预览资料，最后提交。

二. 后台管理

1. 接受或拒绝客人订单

2. 修改酒店信息

3. 处理投诉

4. 发布优惠信息

网站管理员需求

一. 管理员行为需求

1. 管理员注册，登录

- 注册

用户名

身份证号

密码

密码确认

管理码（由公司发放，需要和身份证号对应，否则不能注册）

- 登录

用户名

密码

2. 管理员资料

- 姓名
- 工号

二. 管理功能需求

1. 审核新酒店信息

注册新酒店需要的资料

酒店名称
酒店地址
经营许可证号
联系人姓名
联系人手机号
联系人身份证号
联系人电子邮箱
酒店电话
酒店 id(系统分配)

2. 管理酒店发布的当日住房信息

- 住房信息

房间类型 (标准间、单人房、大床房、商务大床房等, 名字不可重复)
房间规格 面积(平米)
空调信息 (无\格力 xxxx 型号 格式如左侧)
电视信息
宽带信息 (无\100M,支持 wifi)
早餐信息
注意事项
特殊要求
备注
房间价格
住房时间

3. 查看订单列表

- 订单内容

用户名 (手机)
酒店 id
订单 id(系统分配)
酒店名称
酒店地址
房型
入住时间
到期时间
预定日期
总价
已/未完成

- 查看方式

支持全部查看
按订单号查看
按酒店号查看
按预定时间查看
按用户名查看

4. 对用户未完成订单管理

- 订单取消(在满足条件内可以直接取消, 否则拒绝请求)

取消订单理由: 服务太差/环境太差/下订单出错/出行计划改变

取消订单备注：用户填写

5. 对用户未完成订单管理

- 订单评论

删除

标记恶意评论

- 订单投诉

用户投诉功能：对某个订单发起投诉，填写投诉内容

管理员管理投诉：给用户和酒店发送回执

6. 投诉管理

- 所有用户发起的投诉总结
- 投诉内容

投诉理由

投诉酒店名称

投诉酒店 id

投诉人账号

- 投诉回执(发送给酒店和用户)

7. 用户管理

- 用户封禁（封禁理由填写）

8. 酒店管理

- 酒店封禁（封禁理由填写）

2.3 Nofunctional Requerements

Documentation

提供网站操作的说明书以方便用户的操作，提供网站安全性相关文件让用户放心使用

Hardware considerations

需要善于进行多线程处理的服务器以及较大的硬盘存储数据库中的内容，需要在多个地区进行镜像网站以保证各地区的连接速度和使用体验

Performance characteristics

系统速度需要得到保证，要能在短时间内对用户做出的操作进行应答，使用户几乎不需等待即可进行下一步操作，数据大小有限制

Error handling and extreme conditions

系统需要对错误能够做出及时的反应，防止数据丢失等严重问题发生

System interfacing

系统的输入可以来自系统外部，输出可以到达系统外部，但都需要按照一定的格式需求

Quality issues

要求系统的稳定性有保障，及时更新数据，遭遇错误时能在短时间内解决并恢复

System Modifications

系统易变性需求 1.设计的整个系统应该具有良好的可维护性和拓展性，功能之间耦合度较低 2.设计过程中采用前后端分离方式，保证前端的修改不影响后端服务变更 3.可变需求控制在前端页面和前后端数据传送方面，严格控制数据库结构不可变更

Physical Environment

物理环境需求 1.服务器选用宿舍闲置的电脑，操作系统为 ubuntu1.4 2.异常情况状态概述：服务器发热温度过高，服务器连接外网网络异常，停电等状况 3.服务器使用云服务器进行热备

Security Issues

安全性问题 1.需要对用户的各类输入进行检测，避免非法数据和缓冲区溢出 2.数据库绑定本机 ip，通过服务程序设置入口 3.用户登录设置安全 cookie，并且限制 cookie 作用域，防止跨域访问 4.对请求响应的 ip 进行记录，限制访问次数 5.服务器端关闭各类无关的对外服务，避免服务攻击

Resources and Management Issues

资源和管理问题 1.服务器数据进行定时备份，数据库视数据量而定，暂定备份时间为 3 到 7 天一次 2.配置备用服务器 3.数据备份记录备份时间，备份数据区域，和备份人员 4.服务器配置修改应该记录修改时间，修改内容，修改人员

2.4 Constaint

浏览器版本在 IE6 以上

2.5.1 Scenarios

顾客事件流

场景名称：预订酒店

正在参与的参与者示例：想去三亚旅游的 Bob

事件流：

1. Bob 最近带着全家（包括妻子 Alice，儿子 Jack(7 岁)，女儿 Rose(1 岁)，小狗 Cat）去三亚旅游，由于之前曾在网上了解到 Booking 网站是一个评价比较高的预定酒店网站，于是决定注册一个账号并预定一家酒店。
2. Bob 进入网站，点击“注册”按钮后进入注册页面，而后在这个页面 Bob 开始填写注册所必须的电子邮箱地址和私人密码，然而 Bob 输入自己常用的 6 位数字密码却发现提示长度不够，Bob 这才注意到密码时要求混用大小写字母以及特殊符号而且长度必须大于 8 位。
3. 修改了密码后 Bob 成功注册了账号，跳转到个人信息补充页面，Bob 按要求写了自

己的名和姓，选择了更倾向于五星级酒店，这时页面提示 Bob 注册成功，并发送了一封邮件提示 Bob 点击完成注册。

4. Bob 通过自己的邮箱激活了 Booking 账号，并利用网站首页搜索引擎开始搜索合适的酒店。Bob 输入了目的地 Sanya 和入住时间 2015 年 10 月 28 日和退房时间 2015 年 11 月 8 日，跳转到了结果页面，酒店按照综合评分从大到小依次排列在页面之上。
5. Bob 查看了几个酒店，发现很多酒店都不支持带宠物，也没有婴儿车，于是他回到上一级菜单，添加了筛选条件“允许携带宠物”和“带有婴儿车”。
6. Bob 经过比较，选择了 Sanya Marriott Yalong Bay Resort & Spa 这家酒店，点击进入之后，映入眼帘的是酒店的多张精美图片，每张都附有对酒店的详细介绍。图片下面则是酒店的各种特色(如 No booking fees, staff speak: Chinese, Russian 等等)，各种优势设施(如 BBQ facilities, Outdoor pool (all year), Garden) Bob 十分满意。
7. 正当 Bob 准备预定这家酒店时，他无意间发现这里面还有其他客人对它的评价，于是便饶有兴致的看了一下，虽然大部分都是对酒店的赞誉，但是有一些游客反应酒店里的工作人员不会说英语，有时会造成一些不必要的麻烦，Bob 心里面便打了退堂鼓，他将这家酒店加入收藏夹，返回到搜索结果页面进行下一步寻找。
8. 没过多久，Bob 又相中一家名叫 Huayu Resort & Spa Yalong Bay Sanya 的酒店，点击进入之后，Bob 在下面的酒店特色中找到了 Huayu Resort & Spa Yalong Bay Sanya's staff speak: Chinese, Russian, English，于是很高兴地打算预定这家酒店。
9. 正当 Bob 刚进入预订页面时，Bob 的上司 Bad Boss 给 Bob 打电话，说最近有可能要加班，休假可能会延期。Bob 这时查看了一下预定提示，发现这家酒店不支持临时免费取消预定，便返回上一级，再次搜索其他酒店。
10. Bob 最终找到了一家条件不错，又支持临时免费取消预定的酒店，进入预定页面开始选择合适的房间型号，Bob 选择了一间家庭房，进入了下一个页面。
11. Bob 按照页面的提示输入了全家人的名字和年龄，在下面的特殊需求一栏里勾选了吸烟室，婴儿床，狗屋，并在最下面的留言处填了“希望我到的时候是会说英语的服务生来接待我”，而后点击 Continue 进入下一个页面。
12. Bob 发现这个酒店支持在线支付和到酒店支付，为了省去繁琐的支付流程他决定在线支付，然而他发现自己没有绑定信用卡，于是他跳转到另一个页面去绑定信用卡。绑定之后 Bob 顺利地进行了在线支付。
13. 两天之后，Bad Boss 给 Bob 打电话，加班如期而至，无奈的 Bob 只能取消这次旅游，他上了 Booking 网，登陆之后点击“我的订单管理”，进入管理页面。在管理页面中，Bob 进入了“取消订单页面”，点击取消按钮。
14. 随后酒店的工作人员打来电话，询问了退订的理由，在 Bob 叙述之后，对方表示对此次 Bob 没有去成而遗憾并希望 Bob 下次有机会还是可以去他们酒店，Bob 顿时对这家酒店心存好感。于是下面的评论区中给了这家酒店很高的评价和五星好评。

酒店管理员事件流

场景名字：合作酒店注册

正在参与的参与者事例：公寓拥有者 Bob

事件流：

- Bob 从网上看到 Booking 网站招募合作酒店，决定注册将自己的酒店成为 Booking 合作酒店。
- Bob 点击“成为合作酒店”按钮后进入注册页面，Bob 开始填写注册信息：希望与 Booking 合作酒店数量为 1；选择住宿类型为公寓；住宿名称有海景公寓；联系人姓

名为 Bob ; Bob 填写了自己的电子邮箱后点击开始注册按钮, 此时进入新的界面, 系统提示注册完成 10%。

- Bob 填写公寓总数为 1 ; 由于没有酒店官网, 所以酒店官网一栏 Bob 没有填写 ; Bob 选择个人独立拥有公寓 ; Bob 选择接受信用卡/借记卡 ; Bob 填写了公寓的详细地址、邮政编码和自己的联系电话 ; 该页面填写完成, Bob 进入公寓详情界面的填写, 此时系统提示注册完成 30%。
- Bob 选择公寓名称为海景-分层 ; 吸烟选项为可吸烟 ; 卧室数量为 6 ; 客厅数量为 4 ; 浴室间数为 6 ; Bob 根据每间卧室的床铺情况选择床铺详情和可入住人数 ; Bob 根据每个客厅情况选择客厅沙发床数和可入住人数 ; 根据 Bob 选择的情况, Booking 网站计算出最多可入住人数为 15 人, Bob 填写 15 人每晚共需要的价格, 并选择入住少于 15 人时, 客人可以享受房价, 网站多出一行选项, 要求填写人均折扣和最低针对人数, Bob 分别填写 20%和 1 ; Bob 点击保存和继续进入下一个界面, 为上传照片界面, 此时系统提示注册完成 50%。
- Bob 这时接到了妻子出车祸的消息赶忙跑到医院, Booking 的注册进度被保存。
- Bob 接妻子回到家中后继续进行酒店注册, 他再次点击“成为合作酒店”按钮, 在注册页面选择继续注册, 输入了之前注册的邮箱的地址, 收到验证消息并确认后, Bob 继续注册。
- Bob 在上传酒店照片的界面上传了 20 张自己精心挑选的自己公寓照片。上传完毕后点击下一步, 进入设置界面, 此时系统提示注册完成 70%。
- Bob 选择接受信用卡/借记卡付款, 选择接受银联和 VISA ; Bob 选择有资质接收外宾 ; Bob 选择客人在入住前 2 天可免费取消, 否则客人需要支付首晚房费 ; Bob 选择客人在预订时需要支付预付款 ; Bob 选择税费为无增值税无城市税 ; Bob 设置发票信息, 选择对账单接受人为 Bob, 接受人的地址与自己的酒店地址一致 ; Bob 选择不通过渠道管理商或供应商管理房价和房量 ; Bob 点击下一步进入合约界面, 此时系统提示注册完成 90%。
- Bob 仔细阅读合约, 确认无误后勾选下列两个选项

A) 我保证此住宿为一家合法经营的住宿场所, 具备所有必要的营业执照和许可, 并可在首次接到要求时出示这些文件。Booking.com B.V.保留验证及核查您在此次注册中提供的各类信息的权利。

B) 我方已阅读、接受并同意条款与协议。

- Bob 点击下一步, 此时他再次核对酒店信息后点击确认, 提示完成注册。

网站管理员事件流

场景名称 : 网站管理员的日常

参与者实例 : 网站管理员 Mike, 用户 Alice

事件流 :

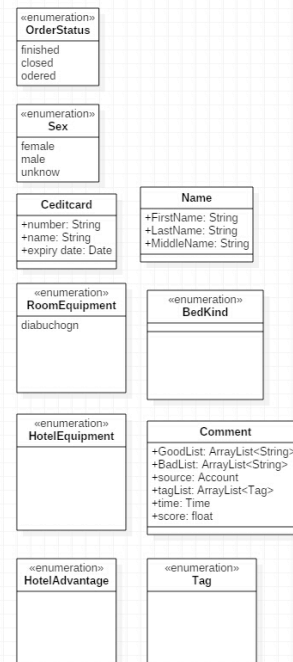
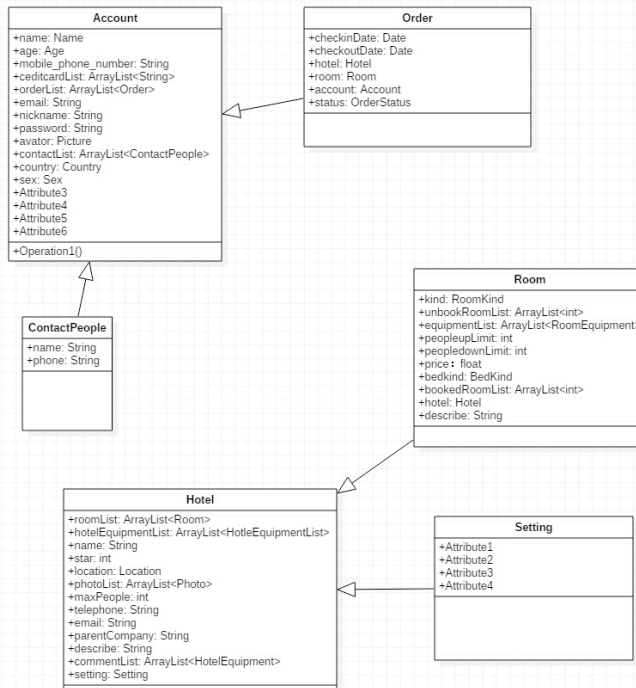
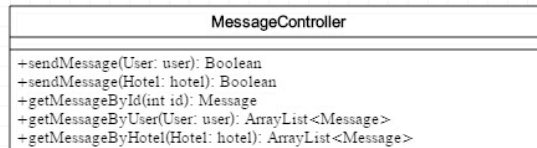
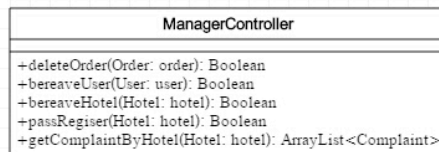
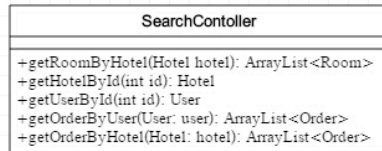
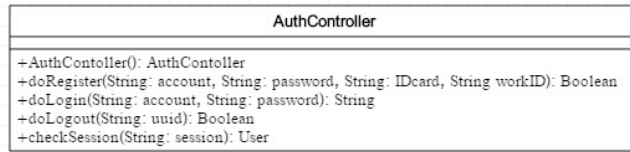
1. Mike 打开电脑, 今天他是第一天上任, 作为 Booking 的网站管理员, 他需要审视一下网站的各类消息。于是他打开网站管理员系统, 但是由于是第一次登录, 他先进行了管理员的注册。Mike 输入了用户名, 身份证号, 密码, 密码确认, 最后输入了公司提供给他的管理码, 这才成功注册。然后他进入登录系统输入用户名和密码进行登陆并且成功。
2. 登录后他又打开了自己的资料页面, 完善了自己的姓名和工号信息。
3. 这时候右上角出现一个红色的数字提示有新的一条消息, 他点开, 原来是本地一家酒店想要和 Booking 公司合作, 他点开合作信息, 新页面是关于酒店的一些资料信

- 息。包括酒店地址、经营许可证号、联系人姓名、联系人手机号、联系人身份证号、联系人电子邮箱、酒店电话。审核完成后，他认为这基本符合了合作要求，就通过了合作请求，然后系统给酒店分配了一个随机 id。
4. 这时候新注册的酒店马上就发布了当日的住房信息，Mike 点开酒店列表，然后进入了这家酒店的页面，页面下有住房、订单两项菜单。点开住房，能看到酒店发布的住房信息以列表形式呈现在眼前，每一条列表都是一类住房的概略信息，包括房间类型，价格和规格。点开列表还可以看到更加详细的信息，房间类型 (标准间、单人房、大床房、商务大床房等)、房间规格 面积(平米)、空调信息、电视信息、宽带信息 (无\100M,支持 wifi)、早餐信息、注意事项、特殊要求、备注、房间价格、住房时间。
 5. 接下来 Mike 又点开了酒店的订单菜单。这家酒店的生意很好，很短时间就已经有好几单订单了，订单内容也是以列表形式呈现在眼前，展示了订单的概略信息，下单的用户名、酒店 id，酒店名，订单 id 和预定日期。订单按照时间先后排序。Mike 使用了搜索功能，搜索条件设定为日期，目标日期为 2015 年 10 月 27 日，马上就出现了今天的所有订单。
 6. Mike 接下来还想看看其他的酒店今天生意如何，就返回到了管理员系统主页，点击了订单管理，然后以列表形式出现了历史的所有订单，Mike 又添加了条件查找，日期为 2015 年 10 月 27 日，酒店名字为自己家周围的某个酒店 A.然后就出现了酒店 A 在这一天的所有订单。
 7. 这时候 Mike 注意到有一个用户 Alice 每天都来这家酒店住，订单记录里每天都有她的名字，于是他返回到了管理员系统主页，点击用户管理，搜索了这个用户，发现她有很多未完成订单，并且今天就下了好几个订单，但是她都选择了取消订单，一看理由都是一些非常荒谬的理由，而且对酒店的每次评论都一样，下单数异常多，于是采取了用户封禁：进入了用户管理页面，点击右侧的用户封禁，使得 Alice 近期内无法再下单，在弹出的回执消息框内填写理由为刷单。并且删除了所有 Alice 异常的评论。
 8. Alice 收到封禁消息后，对酒店 A 进行了投诉，理由是酒店 A 盗用了她的个人信息猜测出了她的密码，并且借她的账号来为自己刷评论。
 9. Mike 发现右上角又有消息提示，于是点开，阅读了 Alice 的投诉请求，然后发送了投诉回执，内容为对 Alice 造成的不便的道歉，然后解封了 Alice 的账号。又查看了酒店 A 最近的订单，有大量此类刷订单的现象，于是向酒店 A 发起了酒店封禁，并且输入了封禁理由为恶意盗用用户隐私，恶意刷评论和订单

2.5.2 Use Case Model

2.5.3 Object Model

Class diagrams



OCL

AuthController 类约束

注册前约束：

context

AuthController::doRegister(String:account,String:password,String:IDcard,String:workID) pre:
!AccountDao::checkRegistered(account)||AccountDao::checkIDcard||AccountDao::checkWord
ID

登录前约束：

context AuthController::doLogin(String:account,String:password) pre:
!@pre.checkSession()||AccountDao::checkRegistered(AccountDao::getAccount(account))

登出约束:

context AuthController::doLogout(String:uuid) post:
!@pre.checkSession()||AccountDao::checkRegistered

SearchController 类约束

context SearchController::getRoomByHotel(Hotel:hotel) inv:
HotelDao.checkHotelsExist(hotel)

context SearchController::getHotelById(int:id) inv:
HotelDao.getHotelById(id)!=null

context SearchController::getUserById(int id) inv:
UserDao.getUserById(id)!=null

context getOrderByUser(User:user) pre:
UserDao.getUserById(user.id)!=null||@pre.checkSession()==user

context getOrderByHotel(Hotel:hotel) pre:
HotelDao.getHotelById(hotel.id)!=null||@pre.checkSession()==hotel.user

ManagerController 类约束

context deleteOrder(Order:order) pre:
@pre.checkSession()==order.user

context bereaveUser(User:user) pre:
AccountDao::isManager(@pre.checkSession())||UserDao::getUserById(user.id)

context bereaveHotel(Hotel:hotel) pre:
AccountDao::isManager(@pre.checkSession())||UserDao::getUserById(hotel.user.id)

context passRegister(Hotel:hotel) pre:
AccountDao::isManager(@pre.checkSession())||HotelDao.getHotelById(hotel.id)

context getComplaintByHotel(Hotel:hotel) pre:
AccountDao::isManager(@pre.checkSession())||HotelDao.getHotelById(hotel.id)

MessageController 类约束

context MessageController::sendMessage(User:user) pre:
UserDao::getUserById(user.id)!=null

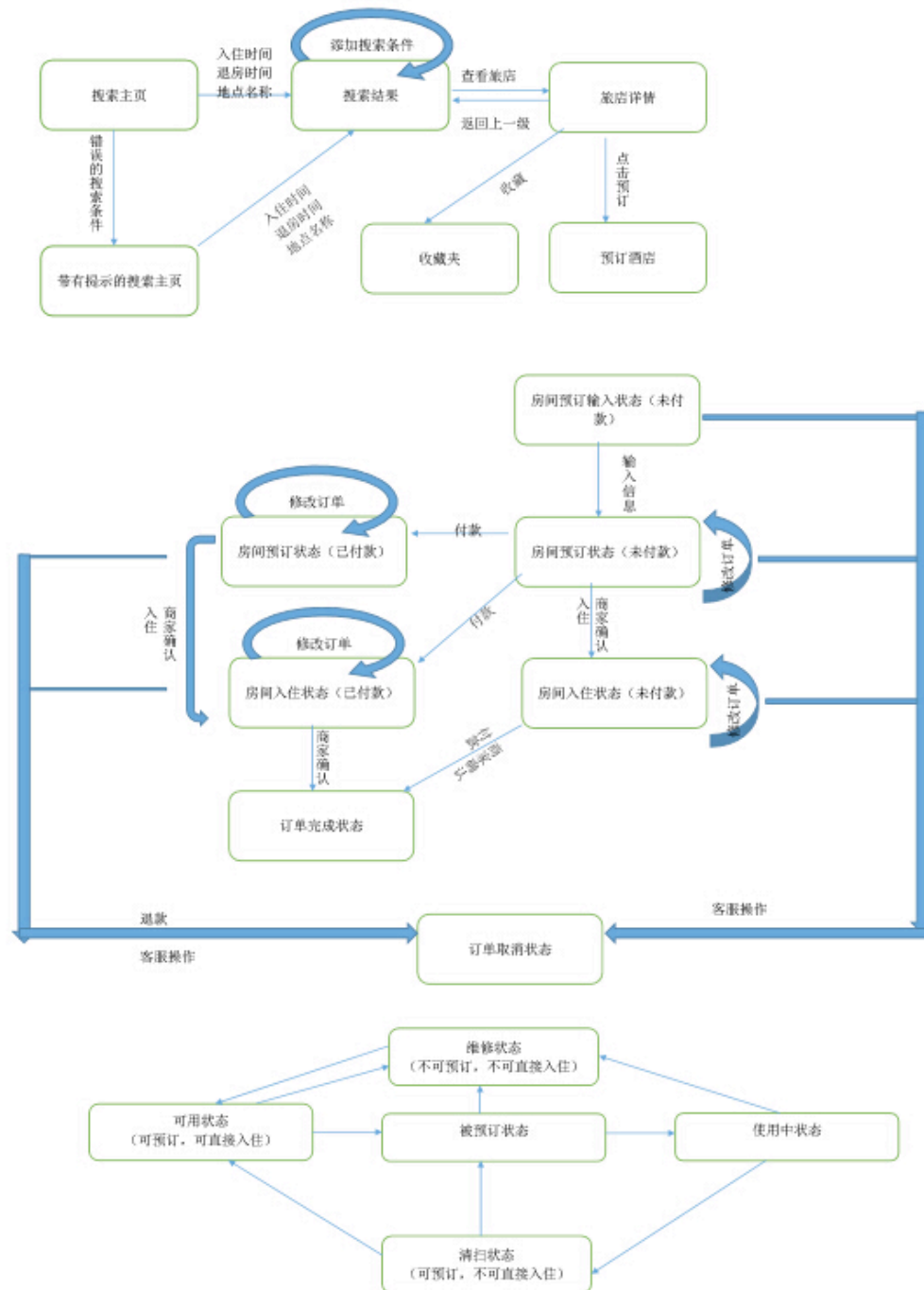
context MessageController::sendMessage(Hotel:hotel) pre:
HotelDao::getHotelById(hotel.id)!=null

context getMessageById(ind id) pre:
AccountDao::isManager(@pre.checkSession())

context getMessageByUser(User:user) pre:
@pre.checkSession().id==user.id||AccountDao::isManager(@pre.checkSession())

context getMessageByHotel(Hotel:hotel) pre:
@pre.checkSession().id==hotel.user.id||AccountDao::isManager(@pre.checkSession())

2.5.4 Dynamic Models



2.5.5 User Interface

3.Glossary