



酒店住宿管理系统 分析与设计

项目组成员：

15331262	饶宇熹	联系方式：13719217597
15331173	李雨珂	联系方式：13719175997
15331281	苏惠玲	联系方式：15521153660
15331216	刘沛东	联系方式：13719176449
15331266	任磊	联系方式：15521221323
15331231	罗淼	联系方式：18073255462
15331270	余滢	联系方式：15768317290

指导老师： 王青

目录

一、需求分析	4
1.1 问题陈述	4
1.2 用例图	5
1.3 用例规约	5
1.3.1 登录系统用例规约	5
1.3.2 管理客房用例规约	7
1.3.3 查询会员用例规约	10
1.3.4 登记入住用例规约	12
1.3.5 预订客房用例规约	15
1.3.6 登记退房用例规约	17
1.3.7 盘点结算用例规约	18
1.3.8 生成财务报表用例规约	19
1.4 补充规约	21
1.5 术语表	21
二、技术选型	22
2.1 数据库 MongoDB	22
2.2 前端 EJS 模板引擎	22
2.3 前端 Semantic UI 框架	22
2.4 后台 Express 框架	23
2.5 日志框架 Winston	23
2.6 加密算法 sha1	23
三、架构设计	23
3.1 架构描述	23
3.2 酒店住宿管理系统架构图	24
3.3 关键抽象	24
四、模块划分	24
4.1 views 页面显示模块	25
4.2 routes 逻辑实现模块	26
4.3 public 页面样式设计模块	27
4.4 lib 自定义通用实体类模块	27
4.5 models 自定义通用函数模块	27
五、类的设计	28

5.1 补充用例规约	28
5.2 用例中类的析取	28
5.2.1 登录系统用例析取	28
5.2.2 管理客房用例析取	30
5.2.3 查询会员用例析取	32
5.2.4 登记入住用例析取	34
5.2.5 预订客房用例析取	39
5.2.6 登记退房用例析取	42
5.2.7 盘点结算用例析取	44
5.2.8 生成财务报表用例析取	46
5.3 分析机制	48
5.4 合并分析类	48
六、子系统及其接口设计	49
6.1 确定设计类	49
6.2 划分子系统	49
七、部件设计	49
7.1 分析并发需求	50
7.2 针对特定需求的设计方案	50
7.2.1 针对住房人数超过酒店限制导致房源不足的解决方案设计	50
7.2.2 针对数据库能否及时更新的解决方案设计	50
八、测试用例设计	51
8.1 等价类划分	51
8.1.1 为登录系统设计等价类	51
8.1.2 为管理客房设计等价类	51
8.1.3 为查询会员设计等价类	52
8.1.4 为预订客房设计等价类	52
8.1.5 为登记入住设计等价类	52
8.1.6 为登记退房设计等价类	52
8.1.7 为生成财务报表设计等价类	53
8.2 边界值分析	53
8.3 测试用例	53
8.3.1 登录系统测试用例	53
8.3.2 管理客房测试用例	54
8.3.3 查询会员测试用例	55
8.3.4 预定客房测试用例	55
8.3.5 登记入住测试用例	57
8.3.6 登记退房测试用例	57
8.3.7 生成财务报表测试用例	57

一、需求分析

1.1 问题陈述

随着我国酒店行业迅速发展，如何在激烈的酒店竞争中脱颖而出，是每个酒店经营者所关注的问题。而 IT 技术为酒店的信息化管理提供了人力操作所不能达到的高效、便利与人性化。传统的酒店住宿管理系统没有保留客户信息，对客户的基本入住习惯也没有基本的了解，很难给人宾至如归的体验。我们设计的酒店住宿管理系统以客户为管理工作的出发点和落脚点，在满足客户的基本居住条件前提下，尽可能使他们住得舒心，真正有如家一般的感觉。

我们开发的酒店住宿管理系统，本着人性化操作的要求设计功能，界面简洁大方，使操作业务一目了然，即使未经过培训的操作人员，只要具备最基本的计算机操作常识就能快速上手。该系统集管理客房、登记入住、预订客房、登记退房、查询会员、盘点结算、生成财务报表等功能于一身。其中，“管理客房”便于有管理员权限的经理对客房进行增加、删除和修改房价的操作。“登记入住”适用于有预订的宾客和未预订的宾客办理入住手续，并简化了办理流程。“预订客房”便于宾客远程订房。每次住房都会赠送相应的积分，通过“查询会员”可获取宾客相关个人信息，包括累计住房积分，当积分达到一定数额时将会获得本酒店限量版纪念品一份。“盘点结算”则是维护本酒店住宿管理系统的必要功能，即是对过期的预订信息进行清除，确保已预订但未入住的宾客占用的客房可得到有效的利用。“生成财务报表”同时也需要管理员权限，获取一段时间的酒店经营获利情况并上传给酒店财务部管理人员。

1.2 用例图

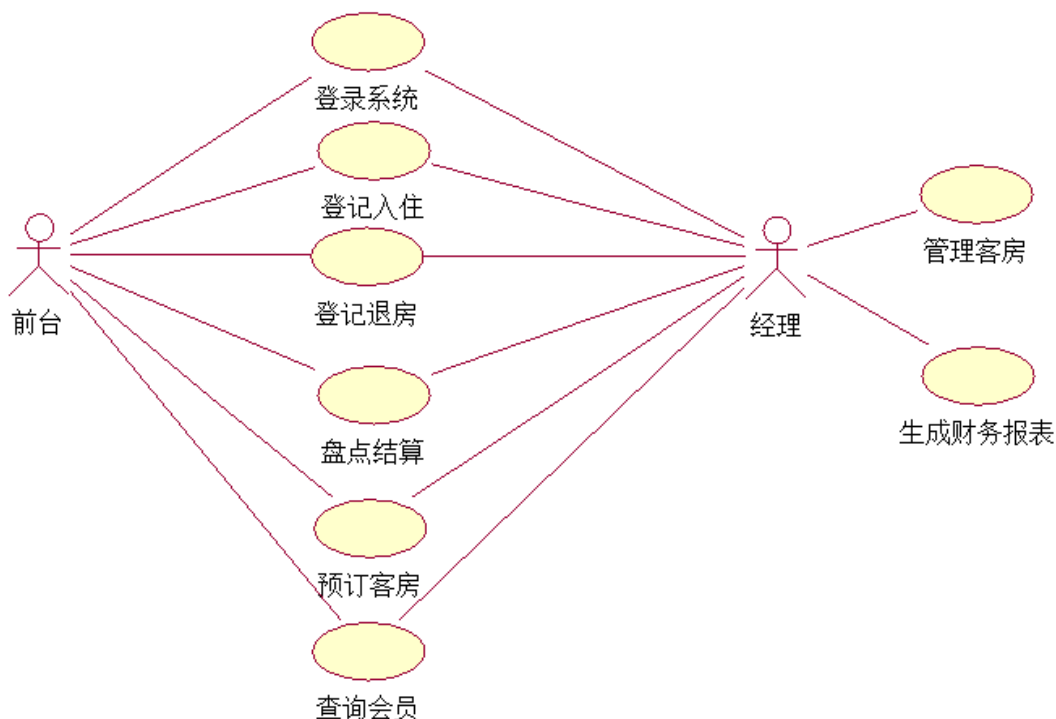


图 1-1 酒店住宿管理系统用例图

1.3 用例规约

1.3.1 登录系统用例规约

1) 简要说明

本用例允许普通前台招待员和酒店经理通过输入用户名和密码来登录酒店住宿管理系统，进入系统功能进而为宾客提供入住相关服务。

2) 事件流

2.1 基本事件流

本用例开始于前台或经理输入相应的信息，进入酒店住宿管理系统。

① 系统首页即为登录页面，如果是前台登录，不需要进行注册操作，直接在“用户名”和“密码”一栏输入相应的信息，点击“登录”即可。

② 如果信息填写正确，系统跳转至酒店管理主页，并显示自当天起 30 天内酒店各种类型房源可用情况列表。

③ 如果信息填写错误，系统提示出错信息，不跳转至酒店管理主页。

④ 如果是经理首次登录，需要完成注册操作。点击页面右上角选择图标进入注册页面，完成“用户名”、“输入密码”、“重复密码”和输入酒店分配的“管理员码”，点击页面下方的“注册”按钮才能成功注册。注册成功之后系统跳转至酒店管理主页，并显示自当天起 30 天内酒店各种类型房源可用情况列表。

2.2 备选事件流

①输入信息有误

C1：用户不存在

- * 当酒店管理人员输入的用户名在数据库中无法找到而点击“登录”时，系统提示“用户不存在”；
- * 系统返回基本事件流第一步。

C2：用户名或密码错误

- * 当酒店管理人员输入的密码与其用户名不匹配而点击“登录”时，系统提示“用户名或密码错误”；
- * 系统返回基本事件流第一步。

②登录信息未填写完整

C1：用户名未填写

- * 当酒店管理人员未填写用户名而点击“登录”时，系统提示“请填写用户名”；
- * 系统返回基本事件流第一步。

C2：密码未填写

- * 当酒店管理人员未填写密码而点击“登录”时，系统提示“请填写密码”；
- * 系统返回基本事件流第一步。

3、活动图

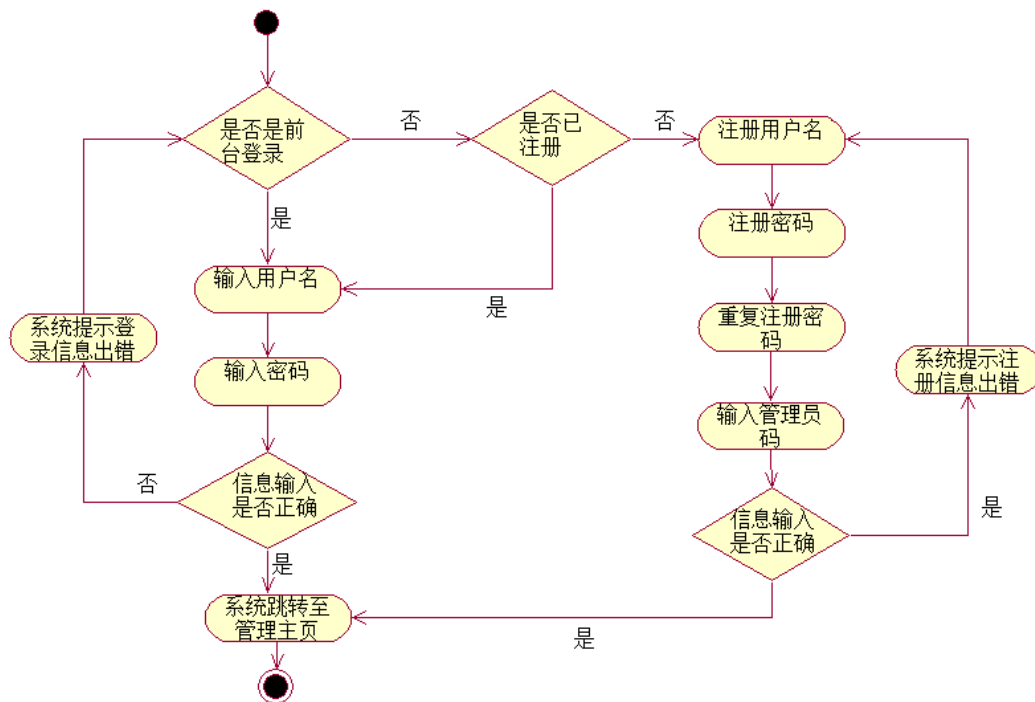


图 1-2 登录系统活动图

4、特殊需求

无

5、前置条件

酒店管理人员打开酒店系统，前台已获取酒店分配的登录用户名和密码。

6、后置条件

酒店管理人员进入住宿管理系统功能主页

1.3.2 管理客房用例规约

1、简要说明

本用例允许具有管理员权限的经理进行增加客房、删除客房、调整房价和重置不规范的房间信息的操作。

2、事件流

2.1 基本事件流

本用例是继“登录系统”之后的一个管理功能，用于客房信息的增、删、改。

① 本用例页面显示四种对房间进行的操作，包括增加客房、删除客房、调整客房房价和重置客房。

② “增加客房”模块包括增加的房间类型、房间号和管理员码。该模块以房间类型为向导，房价类型决定房间价格，在输入房间号和房间类型后，输入正确的管理员码，点击“添加”保存操作信息。

③ “删除客房”模块包括要删除的房间号和管理员码。输入正确的房间号和管理员码，点击“删除”保存操作信息。

④ “调整房价”模块包括起始时间、结束时间、房间类型、房间价格和管理员码。该模块以房间类型为向导，先选择要调整房价的房间类型，输入修改后的房价。输入正确的管理员码，点击“修改”保存操作信息。

⑤ “重置客房”模块是对增加的不规范的房间信息进行清空操作。

2.2 备选事件流

① 增加客房

C1：请填写房间号：数字

当在“房间号”一栏中未填写正确房间信息时报错

C2：房间类型填写错误，正确格式为：单人房/双人房/大房

当在“房间类型”一栏中未填写正确的房间信息时报错

C3：管理员码错误

当在“管理员码”一栏中未填写正确的密码时报错

C4：房间号已被占满

当输入的房间号已存在时会报错

C5：添加失败

当写入数据库失败时报错

② 删除客房

C1：请填写房间号：数字

当在“房间号”一栏中未填写正确房间信息时报错

C2：不存在房间

当输入的房间号不存在时报错

C3：管理员码错误

当在“管理员码”一栏中未填写正确的密码时报错

C4：删除失败

当写入数据库失败时报错

③ 调整房价

C1：起始时间格式错误！正确格式为：（8 位阿拉伯数字表示）YYYYMMDD

当在“起始时间”一栏中未填写正确时间信息时报错

C2：结束时间格式错误！正确格式为：（8 位阿拉伯数字表示）YYYYMMDD

当在“终止时间”一栏中未填写正确时间信息时报错

C3：结束时间不能早于起始时间

当结束时间早于起始时间时报错

C4：起始时间不能早于今日

即日起修改房价，无法对以往房间房价进行修改

C5：房间类型填写错误，正确格式为：单人房/双人房/大房

当在“房间类型”一栏中未填写正确的房间信息时报错

C6：房间价格填写有误

当填入信息低于设定值时报错

C7：管理员码错误

当在“管理员码”一栏中未填写正确的密码时报错

C8：修改失败：超出 30 天限制

只能对一个月内的房间价格进行调整

3、活动图

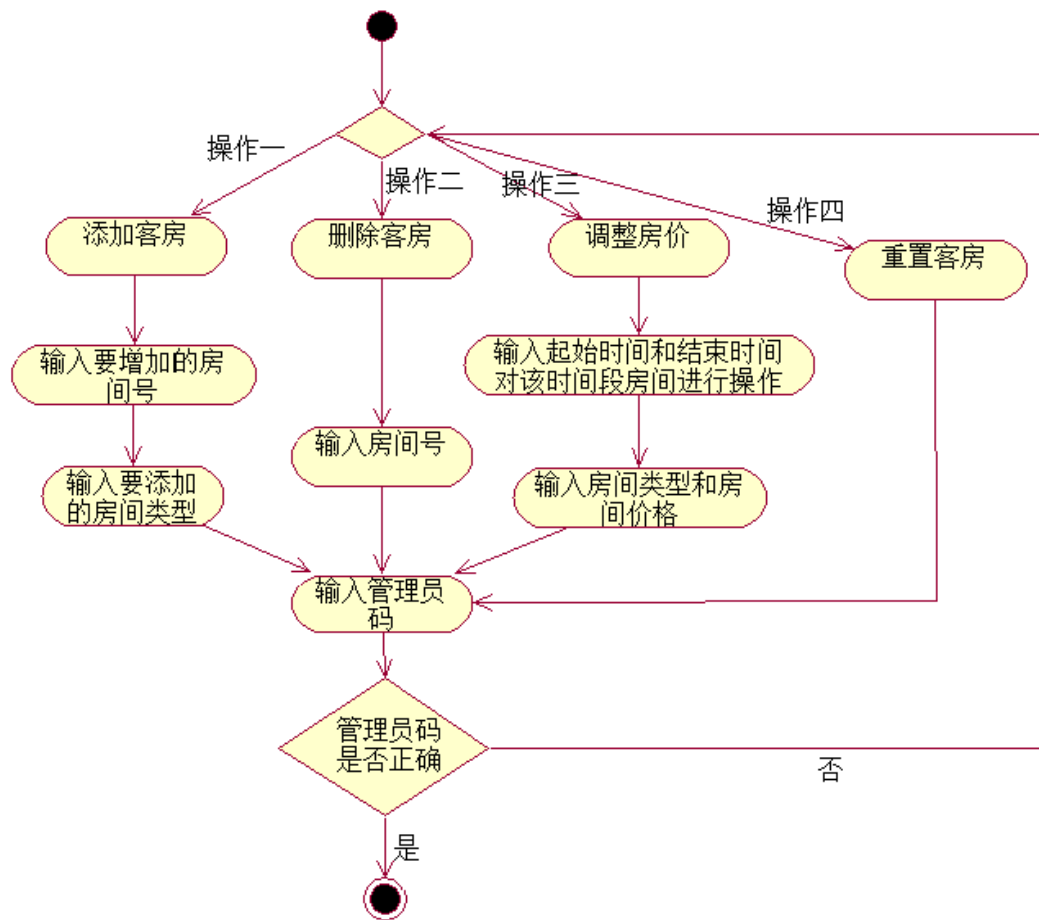


图 1-3 管理客房活动图

4、特殊需求

无

5、前置条件

酒店管理人员已完成系统登录, 进入系统管理主页, 且只有经理有管理权限。

6、后置条件

系统管理主页显示自登录日起 30 天内酒店各种类型房源可用情况列表随着对客房信息的操作实时进行更新。

1.3.3 查询会员用例规约

1、简要说明

本用例用于查询会员基本信息, 获取会员积分状况。

2、事件流

2.1 基本事件流

本用例是继“登录系统”之后的一个管理功能，用于查询会员的基本信息。

- ① 本用例需要先输入会员的身份证号进行搜索
- ② 若能查找到该会员，则显示会员的姓名、联系电话、身份证号、积分。
- ③ 若未查找到该会员，则提示“会员不存在”

2.2 备选事件流

① 会员不存在

说明该宾客是第一次入住本酒店，故系统上没有保存个人信息，提示“会员不存在”。有过至少一次入住登记的宾客则为酒店会员，系统上能显示其个人信息。

② 无效身份证号

当输入的身份证号长度低于 18 位时报错

3、活动图

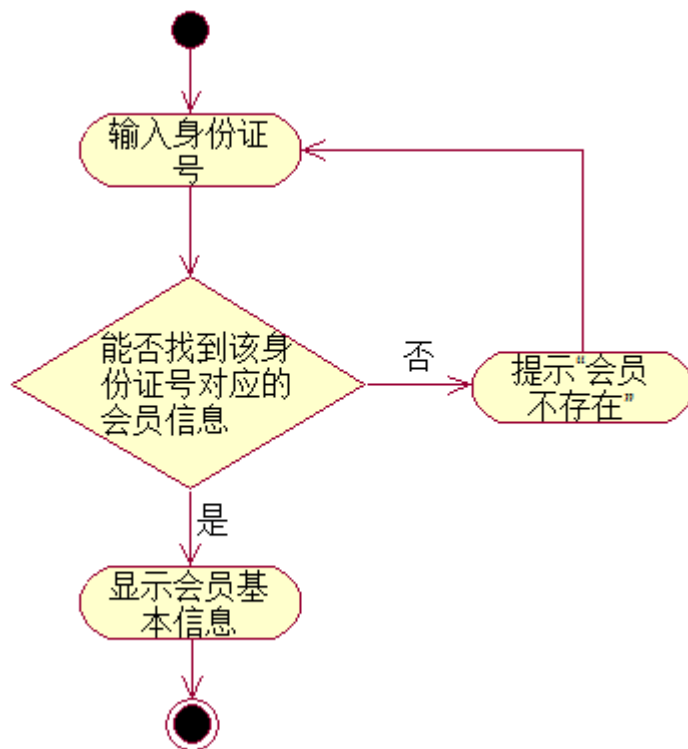


图 1-4 查询会员活动图

4、特殊需求

无

5、前置条件

酒店管理人员获取宾客身份证号。

6、后置条件

如果能查找到会员信息，根据会员积分情况赠予酒店限量版纪念品。

1.3.4 登记入住用例规约

1、简要说明

本用例允许前台或经理为已预订和未预订的宾客办理入住手续，其中，身份证号是系统判别入住宾客是否预订的唯一标识。

2、事件流

2.1 基本事件流

本用例是继“登录系统”之后的一个管理功能，用于办理入住手续。

① 前台或经理输入入住宾客的身份证号，分别进行预定查询和会员查询。

② 点击“预定查询”，如果是已预订的宾客，则显示所有预订信息。预订信息包括个人信息和房间信息，个人信息包括预订人姓名、联系方式、身份证号；房间信息包括房间类型、入住时间和退房时间。

③ 选择一条预定信息，点击“获取房间号”。系统在可用房源中随机选择一个符合预订信息的房间，在获取房间号后，前台或经理点击对话框中的“确认入住”即完成已预订宾客入住手续的办理。

④ 如果是未预订的宾客，查询结果返回对话框提示“未预订”。

⑤ 点击“会员查询”，如果是酒店会员，系统会显示该会员个人信息。否则需要手动输入宾客的姓名、联系方式和身份证号。

⑥ 填写完个人信息后，前台或经理在“房间类型”、“入住时间”和“退房时间”栏中输入宾客要求的房间信息。点击“获取房间号”，系统在可用房源中随机选择一个符合预订信息的房间，在获取房间号后，前台或经理点击对话框中的“确认入住”即完成未预订宾客入住手续的办理。

2.2 备选事件流

① 有预订的顾客进行登记入住操作

C1：无效身份证号

当输入的身份证号位数不为 18 位时报错

② 无预订的会员进行登记入住操作

C1: 无效身份证号

当输入的身份证号位数不为 18 位时报错

C2: 预订信息不存在

当未预订时点击“预订查找”时报错

C3: 没有当天入住的预订

当该会员有预订但不是预订当天入住时会报错

C4: 该房型无空房

未预订的顾客入住可能会遇到没有房间的问题

③ 无预订的普通顾客进行登记入住操作

C1: 无效身份证号

当输入的身份证号位数不为 18 位时报错

C2: 非会员

当该顾客点击“会员查找”时会报错

C3: 姓名不能为空

当未输入顾客的姓名时报错

C4: 无效手机号

当输入的手机号非 11 位或者输入的非数字时会报错

C5: 房间类型填写错误, 正确格式为: 单人房/双人房/大房

当在“房间类型”一栏中未填写正确的房间信息时报错

C6: 退房时间格式错误! 正确格式为: (8 位阿拉伯数字表示) YYYYMMDD

当在“退房时间”一栏中未填写正确时间信息时报错

C7: 退房时间不能早于入住时间

当退房时间早于入住时间时报错

C8: 该房型无空房

未预订的顾客入住可能会遇到没有房间的问题

3、活动图

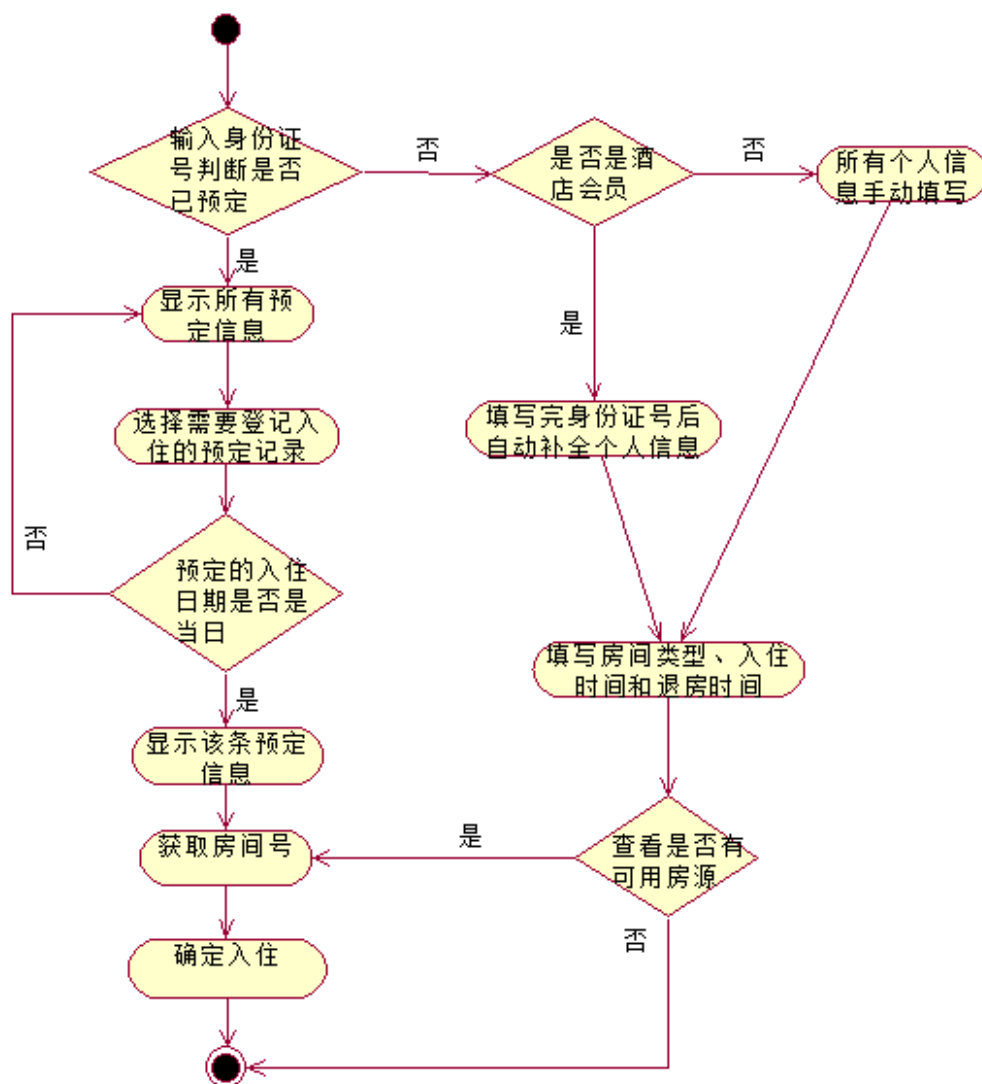


图 1-5 登记入住活动图

4、特殊需求

无

5、前置条件

如果宾客想通过预订入住，需提前与前台或经理联系完成预订手续，入住时只需凭借预订时的身份证号即可顺利入住。如果是没有预订的宾客，可能面临没有可用房源的情况。

6、后置条件

入住手续办理成功，宾客需要在退房时间到之前到酒店前台完成退房手续。

1.3.5 预订客房用例规约

1、简要说明

本用例允许前台或经理远程为宾客查找是否有符合宾客入住条件的可用房源，如果有则会为宾客预留至规定入住时间。

2、事件流

2.1 基本事件流

本用例是继“登录系统”之后的一个管理功能，用为宾客预订客房。

① 当接到宾客想预订客房的电话时，前台或经理在系统主页查看自当天起30天内酒店各种类型房源可用情况列表，询问宾客入住时间、退房时间和房间类型。

② 如果有可用房源，进入“预订客房”模块进行预订信息登记。预订信息包括个人信息和房间信息。

③ 个人信息包括身份证号、姓名和联系电话。在个人信息中先输入身份证号，系统判定该宾客是否是会员。如果是，其余个人信息会自动生成；如果不是，其余个人信息需手动填写。

④ 房间信息包括房间类型、入住时间和退房时间，房间信息需要手动填写。

⑤ 点击屏幕下方的“预订客房”按钮即可完成预订。

2.2 备选事件流

① 该会员不存在

入住顾客是首次入住，系统数据库没有其个人信息

② 无效身份证号

当输入的身份证号位数不为18位时报错

③ 房间类型填写错误，正确格式为：单人房/双人房/大房

当在“房间类型”一栏中未填写正确的房间信息时报错

④ 没有足够的房间

当没有足够的房间可以预订时会报错

⑤ 退房时间格式错误！正确格式为：（8位阿拉伯数字表示）YYYYMMDD

当在“退房时间”一栏中未填写正确时间信息时报错

⑥ 入住时间格式错误！正确格式为：（8 位阿拉伯数字表示）YYYYMMDD

当在“入住时间”一栏中未填写正确时间信息时报错

⑦ 退房时间不能早于入住时间

当退房时间早于入住时间时报错

⑧ 预订时间不能早于今日

当预订时间早于今日时报错

⑨ 不能登记超过 30 天的预订

预订时间限制在 30 天内，超出会报错

3、活动图

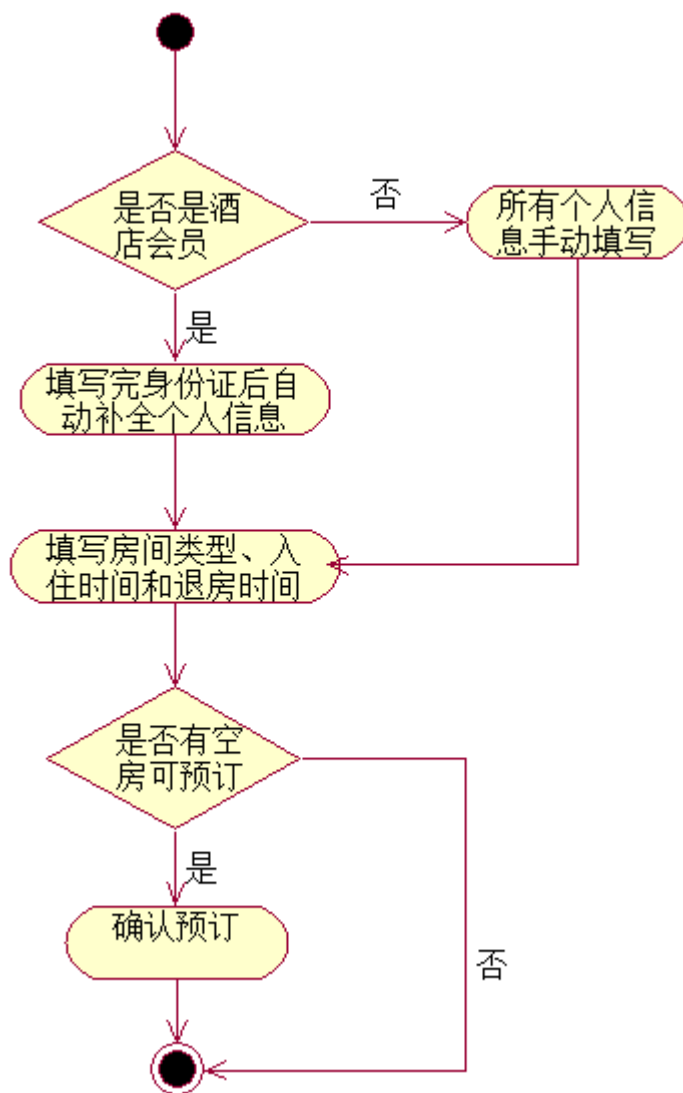


图 1-6 预订客房活动图

4、特殊需求

对于预订的宾客，不得迟于填写的入住时间两小时，否则系统自动删除预订信息，此次预订失效。

5、前置条件

前台或经理获取宾客的入住时间、退房时间和房间类型，在自当天起 30 天内酒店各种类型房源可用情况列表中查看到有可用房源。

6、后置条件

完成线上预订的宾客需要在规定的入住时间之前携带身份证到酒店办理登记入住手续。

1.3.6 登记退房用例规约

1、简要说明

本用例用于为住房时间到达退房时间的宾客快速办理退房手续。

2、事件流

2.1 基本事件流

① 酒店管理人员获取宾客的房间号，搜索得到该房间的入住信息。

② 给宾客确认入住信息是否有误，如果没有，直接点击屏幕下方的“退房”

即可完成退房手续。如果入住信息有误，再与宾客进一步沟通。

2.2 备选事件流

① 房间号不存在

系统提示输入的房间号有误

3、活动图

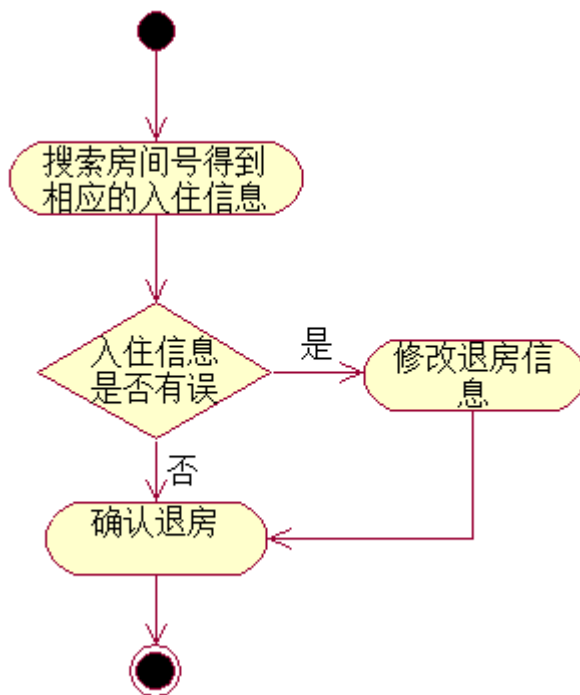


图 1-7 登记退房活动图

4、特殊需求

对于退房的宾客，办理退房手续的时间不得迟于规定退房时间后两个小时，否则加收管理费。

5、前置条件

宾客已完成“登记入住”手续办理。

6、后置条件

前台或经理核对该房间原置非赠送用品是否损坏。如果没有损坏，宾客即可离开酒店。如果有损坏，向宾客加收物品损坏费，然后宾客才可离开酒店。

1.3.7 盘点结算用例规约

1、简要说明

本用例用于删除过期的预订信息，并更新可用房源信息。

2、事件流

2.1 基本事件流

① 前台或经理进入“盘点结算”板块，点击“结算当日信息”删除现存的过期的预订信息。

2.2 备选事件流

暂无

3、活动图

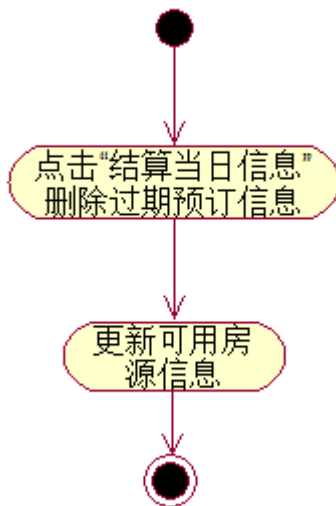


图 1-8 盘点结算活动图

4、特殊需求

无

5、前置条件

无

6、后置条件

删除现存的过期的预订信息后页面返回管理系统的主页。

1.3.8 生成财务报表用例规约

1、简要说明

本用例用于统计酒店一段时间的住宿盈利情况，将住宿信息汇总后交给财务系统。该操作只有经理有权限，普通前台没有权限查看财务情况。

2、事件流

2.1 基本事件流

- ① 进入“生成财务报表”功能，输入管理员码确定操作人有经理权限。
- ② 经理滑动时间滑块选择要统计的住宿盈利情况时间段。点击财务报表下

载按钮，报表以 excel 格式生成。

2.2 备选事件流

① 请填写开始时间

系统提示未填写要下载的入住信息的开始时间

② 请填写结束时间

系统提示未填写要下载的入住信息的开始时间

③ 结束时间格式错误！正确格式为：（8 位阿拉伯数字表示）YYYYMMDD

当在“结束时间”一栏中未填写正确时间信息时报错

④ 开始时间格式错误！正确格式为：（8 位阿拉伯数字表示）YYYYMMDD

当在“开始时间”一栏中未填写正确时间信息时报错

⑤ 管理员码错误

当输入的管理员码错误时报错

3、活动图

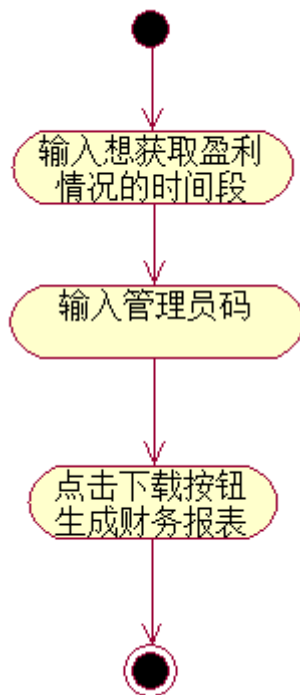


图 1-9 生成财务报表管理活动图

4、特殊需求

无

5、前置条件

经理才有权限获取酒店一段时间的盈利情况。

6、后置条件

经理将财务报表发给该公司的财务部进行财政数据的录入统计。

1.4 补充规约

1.4.1 目录

本文档主要定义了酒店住宿管理系统的需求，本补充规约流出了不便于在用例模型中获取的系统需求，补充规约和用例模型一并记录了系统需求。

1.4.2 范围

- ① 本补充规约“适用于酒店住宿管理系统”网站
- ② 本规约定义了一些功能性需求和非功能性需求，如可靠性、可用性

1.4.3 参考

无

1.4.4 可行性

本系统为网站平台，需要在有网络环境中才能运行。

1.4.5 可靠性

本网络系统提供 7*24 小时不间断服务

1.4.6 安全性

- ① 只有酒店管理人员才能访问
- ② 前台管理人员有登录系统、登记入住、预订客房、登记退房、查询会员、盘点结算等功能的权限，但没有管理客房信息和获取财务报表的权限。
- ③ 经理除了享有与前台一样的管理功能外，还有对客房信息进行管理（包括添加客房、删除客房和修改房价的功能）和获取一段时间财务报表的权限。

1.4.7 可支持性

网站系统支持在不同浏览器中运行

1.5 术语表

表 1-10 术语表

名词术语	描述
前台	有登录系统、登记入住、预订客房、登记退房、查询会员、盘点结算等功能的权限。
经理	有登录系统、登记入住、预订客房、登记退房、查询会员、盘点结算、管理客房和获取财务报表等功能的权限。
身份证号	身份证号作为是否预订的唯一标识，也是宾客是否是酒店会员的唯一标识。
会员	首次入住宾客非酒店会员，在关于个人信息的输入中，输入身份证号后进行查询，无法获取其他个人信息。有过入住记录的宾客为酒店会员，在关于个人信息的输入中，输入身份证号后进行查询，其他个人信息会自动补全。
财务报表	财务报表反映酒店一段时间的盈利情况
积分	不同类型的房间由于价格不同会有不同的积分，宾客的积分累计到一定数值可兑换酒店限量版纪念品。

二、技术选型

2.1 数据库 MongoDB

MongoDB 是一个基于分布式文件存储的数据库，它的特点是高性能、易部署、易使用，存储数据非常方便，具有模式自由、支持动态查询、支持完全索引、可通过网络访问等功能。mongoDB 提供的 connect-mongo 工具可实现将 session 持久化保存到 mongoDB。

2.2 前端 EJS 模板引擎

EJS 是一套简单的模板语言，利用 JavaScript 代码生成 HTML 页面。优点在于它是一门友好的编程语言，语法简单，支持在直接在标签内书写简单、直白的 JavaScript 代码，适用于快速开发，无需浪费时间钻研那些所谓“优雅”的神秘语法，也不用研究数据究竟如何能够被正确处理。除此之外，EJS 能够缓存 JS 函数的中间代码，从而提升执行速度。并且，当错误发生时，调试 EJS 错误很容易，因为所有错误都是普通的 JavaScript 异常，并且还能输出异常发生的位置。

2.3 前端 Semantic UI 框架

Semantic UI 集成了很多漂亮的 UI 模板，能够使网页制作更加高效和美观，而且可以通过字面的语义，比如 segment、item、ui button 等来自定义网页，适用于快速开发。

2.4 后台 Express 框架

Express 是一个简洁而灵活的 node.js Web 应用框架，提供了一系列强大特性帮助我们创建各种 Web 应用，和丰富的 HTTP 工具。使用 Express 可以快速地搭建一个完整功能的网站，通过设置 session 中间件来相应 HTTP 请求，定义了路由表用于执行不同的 HTTP 请求动作，并且可以通过向模板传递参数来动态渲染 HTML 页面。本系统主要用到 express-session 中间件来响应 HTTP 请求，connect-flash 中间件实现一次性的信息提示，express-formidable 中间件用于处理表单和上传文件。

2.5 日志框架 Winston

Winston 是一个简单通用的日志库，日志格式容易被人和机器解读，支持多传输（在 Winston 中，一个传输实质上代表储存设备，也就是数据最终保存在哪里），每个 Winston 实例都可以对不同级别的日志配置不同的传输。

2.6 加密算法 sha1

安全哈希算法（Secure Hash Algorithm）主要适用于数字签名标准（Digital Signature Standard DSS）里面定义的数字签名算法（Digital Signature Algorithm DSA）。对于长度小于 2^{64} 位的消息，SHA1 会产生一个 160 位的消息摘要。当接收到消息的时候，这个消息摘要可以用来验证数据的完整性。在传输的过程中，数据很可能会发生变化，那么这时候就会产生不同的消息摘要。本系统使用该算法用于登陆系统的用户密码加密。

三、架构设计

3.1 架构描述

本系统将采用 MVC（Model View Controller）架构设计。该架构的使用实现了应用程序的分层管理，简化了后续对程序的修改和扩展，并且使程序某一部分的重复利用成为可能。

View 层又称视图层，是用户看到并与之交互的页面。在本系统中，视图层由众多 HTML 文件组成，它们负责获取用户的输入及显示控制层处理的结果。

Controller 层又称控制层，它收到来自视图层的请求并对实体类进行增删改查。当单击 Web 页面中的超链接和发送 HTML 表单时，控制层接收请求并处理请求，然后确定用哪个视图来显示处理返回的数据。

Model 层又称数据层，主要作用是存储和访问数据。

3.2 酒店住宿管理系统架构图

经分析，本研究平台的架构如图 3-1 所示：

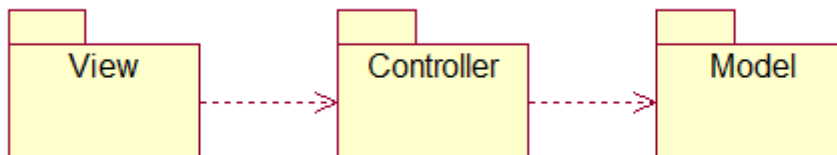


图 3-1 酒店住宿管理系统架构图

3.3 关键抽象

经分析，本系统有 6 个实体类，分别为用户类 (User)，顾客类 (Guest)，房间类 (Room)，预订信息类 (BookInfo)，入住信息类 (Checkin)，剩余房间类 (AvailableRoom)。其中，用户类存储了登录该系统所需要的用户名和密码。顾客类存储了与住宿宾客有关的信息，包括姓名、身份证号、联系方式和积分等。房间类存储酒店房间有关的信息，包括房间号、房间类型、房间价格和房间状态。预订信息类存储了预订客房的相关信息，包括预订人的身份证号、联系方式、姓名，预订的房间类型，预订入住的起始时间和结束时间。入住信息类存储了入住客房的相关信息，包括入住人的身份证号、姓名、联系方式，入住房间的房间号，入住时间和退房时间。剩余房间类存储了当日起 30 天内可入住的房间信息，包括年月日基本时间信息，各种房间类型（大床房、单人房和双人房）可入住的数量。

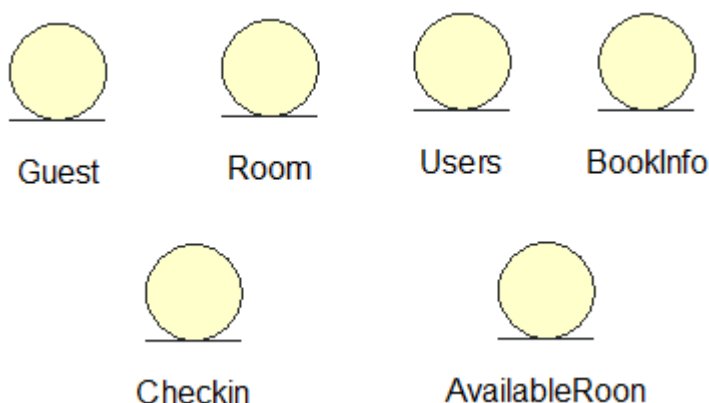
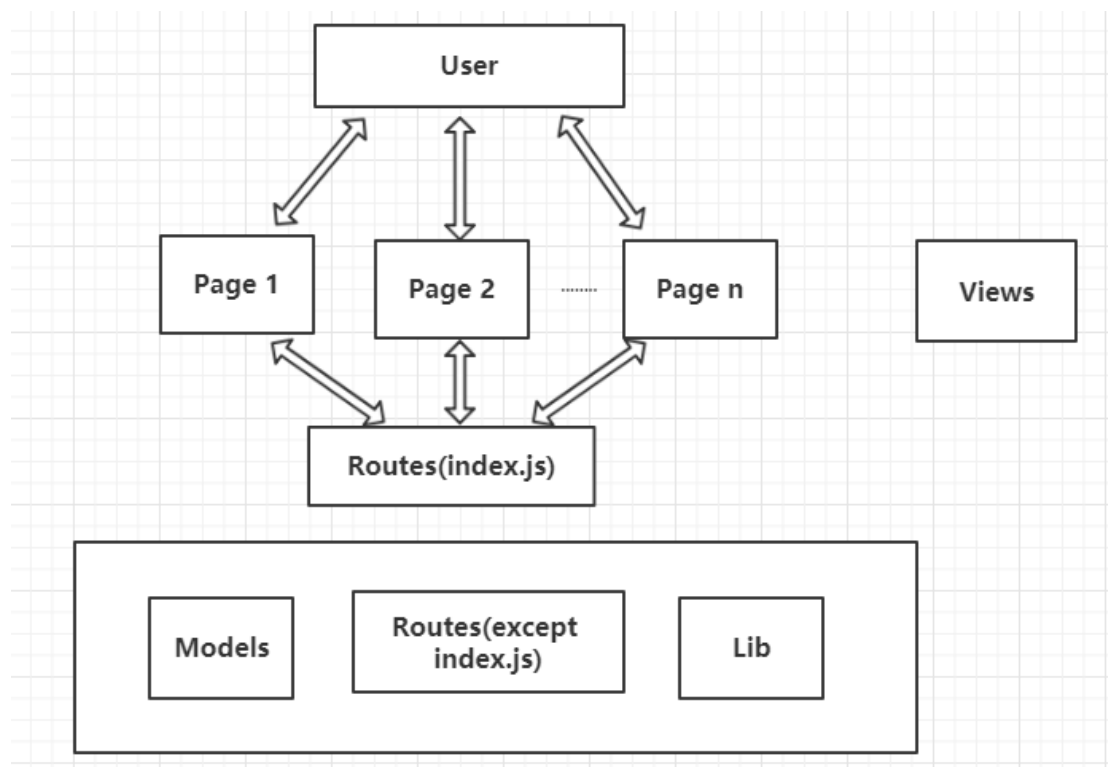


图 3-2 酒店住宿管理系统实体类图

四、模块划分

根据 MVC 架构，本系统对页面元素进行了具有层次结构的划分，主要模块有 views 页面显示模块，routes 逻辑实现模块（其中包含了路由），public 页面样

式设计模块，lib 自定义通用实体类模块和 models 自定义通用函数模块。除此之外，还单独分出了 config 读取配置文件模块，middleware 中间件模块。



4.1 views 页面显示模块

views 页面显示模块包括单独的 ejs 文件页面和 components 文件，放在 /views 目录下。

```

└─ views
  └─ components
    <> 404.ejs
    <> addroom.ejs
    <> balance.ejs
    <> bookroom.ejs
    <> checkin.ejs
    <> checkout.ejs
    <> deleteroom.ejs
    <> finance.ejs
    <> footer.ejs
    <> footerbr.ejs
    <> footerci.ejs
    <> footerco.ejs
    <> footermr.ejs
    <> header.ejs
    <> manage.ejs
    <> manageroom.ejs
    <> resetroom.ejs
    <> searchcus.ejs
    <> signin.ejs
    <> signup.ejs
    <> updateroom.ejs

```

- ✧ 404.ejs: 显示页面已丢失
- ✧ addroom.ejs: 添加客房页面
- ✧ balance.ejs: 盘点结算页面
- ✧ bookroom.ejs: 预定客房页面
- ✧ checkin.ejs: 登记入住页面
- ✧ checkout.ejs: 登记退房页面
- ✧ deleteroom.ejs: 删除客房页面
- ✧ finance.ejs: 获取财务报表页面
- ✧ footer.ejs: 定义 section 的页脚
- ✧ footerbr.ejs: 定义 section 的页脚
- ✧ footerci.ejs: 定义 section 的页脚
- ✧ footerco.ejs: 定义 section 的页脚
- ✧ footermr.ejs: 定义 section 的页脚
- ✧ header.ejs: 定义 section 的页眉
- ✧ manage.ejs: 系统主页
- ✧ manageroom.ejs: 客房管理操作主页
- ✧ resetroom.ejs: 重置客房主页
- ✧ searchcus.ejs: 查询会员信息主页
- ✧ signin.ejs: 登录主页
- ✧ signup.ejs: 注册主页
- ✧ updateroom.ejs: 修改客房主页

4.2 routes 逻辑实现模块

```

└─ routes
  JS balanceclear.js
  JS bookroom.js
  JS checkin.js
  JS checkout.js
  JS finance.js
  JS index.js
  JS manage.js
  JS manageroom.js
  JS searchcus.js
  JS searchroom.js
  JS signin.js
  JS signout.js
  JS signup.js

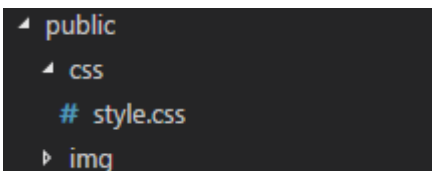
```

- ✧ balanceclear.js: 实现与盘点结算有关的逻辑
- ✧ bookroom.js: 实现与预定客房有关的逻辑
- ✧ checkinroom.js: 实现与登记入住有关的逻辑
- ✧ checkoutroom.js: 实现与登记退房有关的逻辑
- ✧ finance.js: 实现与生成财务报表有关的逻辑
- ✧ index.js: 定义前端与后台交互的接口
- ✧ manage.js: 实现系统主页的逻辑
- ✧ manageroom.js: 实现管理客房的逻辑, 其中包括增加客房、删除客房、调整房价和重置客房
- ✧ searchcus.js: 实现查询会员的逻辑
- ✧ signin.js: 实现登录系统的逻辑
- ✧ signout.js: 实现登出系统的逻辑
- ✧ signup.js: 实现系统上注册的逻辑

```
// manage
app.get('/manage', checkLogin, function(req, res) {
  |   manageRouter["managePage"](req, res);
})
app.post('/manage', checkLogin, function(req, res) {
  |   manageRouter["clear"](req, res);
})
```

在 index.js 文件中，定义了前端通过 GET 方法从指定的服务器中获取数据，通过 POST 方法提交请求的参数给指定的服务器处理。后台通过 require() 方法引入需要的模块，并接收到前端传入的参数，便于在数据库中实现查询，查询完毕将结果返回给前端。

4.3 public 页面样式设计模块



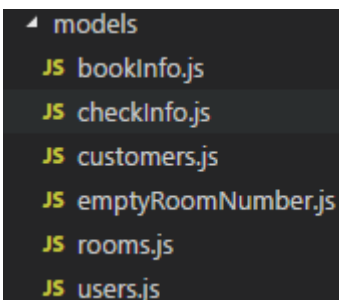
```
└─ public
  └─ css
    # style.css
  └─ img
```

这一模块包括对系统各页面 css 样式的设计和系统设计所用到的图片

4.4 lib 自定义通用实体类模块

此模块自定义系统所用的 6 个实体类，分别是分别为用户类(User)，顾客类(Guest)，房间类(Room)，预订信息类(BookInfo)，入住信息类(Checkin)，剩余房间类(AvailableRoom)。关于实体类的具体介绍在 3.3 关键抽象中已提及。

4.5 models 自定义通用函数模块



```
└─ models
  JS bookInfo.js
  JS checkInfo.js
  JS customers.js
  JS emptyRoomNumber.js
  JS rooms.js
  JS users.js
```

此模块自定义系统的通用函数，比如会员信息查询相关的函数并不限于查询会员信息的页面，在登记入住和预定客房的页面也需要对会员信息进行查询。这些模块独立起来方便复用，为不同页面提供支持，体现了结构化编程的思想。

五、类的设计

5.1 补充用例规约

经检查，本项目组发现用例规约比较完善，暂时无需补充。

5.2 用例中类的析取

5.2.1 登录系统用例析取

根据第一章系统需求分析中的用例规约，经分析可得该用例中的三种类。

- 边界类：LoginSystemViewView。LoginSystemViewView 边界类为系统用户登录该酒店住宿管理系统。
- 控制类：LoginController。LoginController 控制类负责处理登录系统的相关操作，包括输入用户名和密码，并对输入的信息进行判断。
- 实体类：User。User 用户类类表示用户实体类，包括用户名和密码。

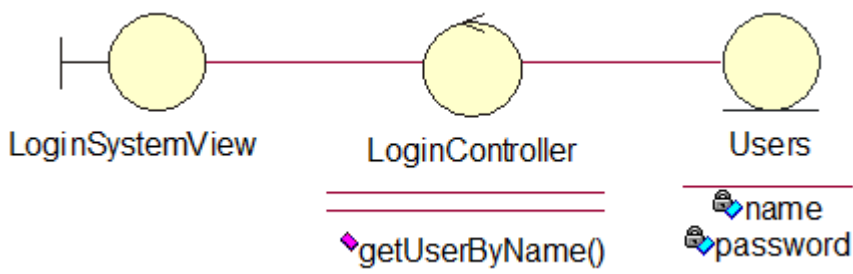


图 5-1-1 登录系统用例析取图

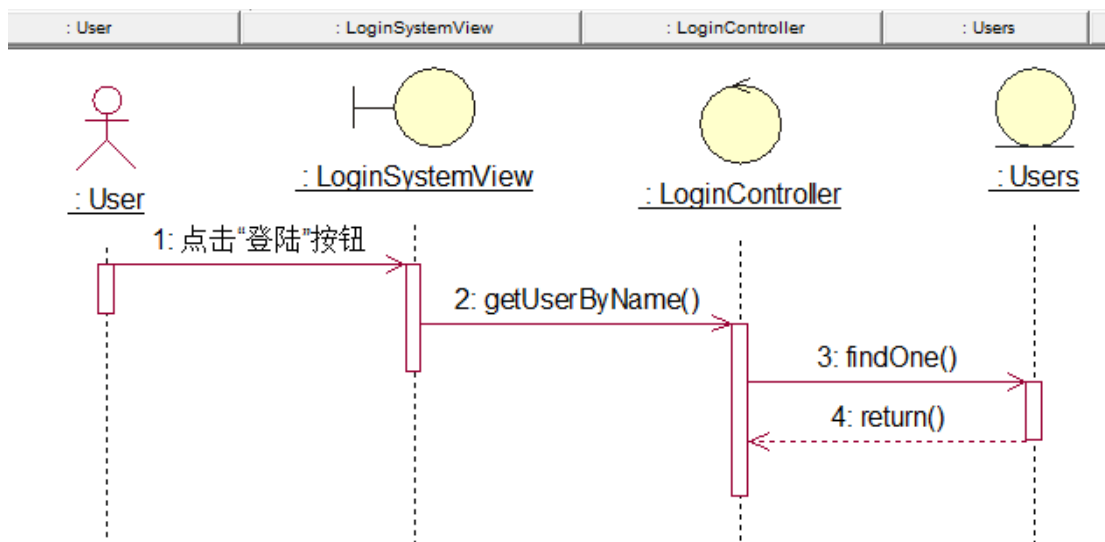


图 5-1-2 登录系统时序图

用户在登录页面在“用户名”和“密码”一栏中分别输入信息，点击“登录”按钮，系统调用控制类的 `getUserByName()` 方法，从输入栏中得到登录的用户名，然后调用实体类 `User` 的 `findOne()` 方法获取对应的用户信息，返回匹配结果。如果填写无误返回“登录成功”并跳转到系统功能主页面，否则返回出错信息。

<> signin.ejs × 该 ejs 文件对应登陆系统的边界类 LoginSystemView

```

1  <%- include('header') %>
2
3  <div class="ui grid">
4    <div class="four wide column"></div>
5    <div class="eight wide column">
6      <form class="ui form segment" method="post">
7        <div class="field required">
8          <label>用户名</label>
9          <input placeholder="用户名" type="text" name="name">
10        </div>
11        <div class="field required">
12          <label>密码</label>
13          <input placeholder="密码" type="password" name="password">
14        </div>
15        <input type="submit" class="ui button fluid" value="登录">
16      </form>
17    </div>
18  </div>
19
20  <%- include('footer') %>
  
```

在登录页面输入“用户名”和“密码”后点击“登录”按钮



5.2.2 管理客房用例析取

根据第一章系统需求分析中的用例规约，经分析可得该用例中的三种类。

- 边界类：ManageRoomView。ManageRoomView 为酒店经理对客房数量和价格进行操作。
- 控制类：ManageRoomViewController。ManageRoomViewController 控制类负责进行酒店客房管理的相关操作，包括增加客房、删除客房、调整房价和重置客房。
- 实体类：Room。Room 房间类包括房间号、房间类型、房间价格和房间状态。

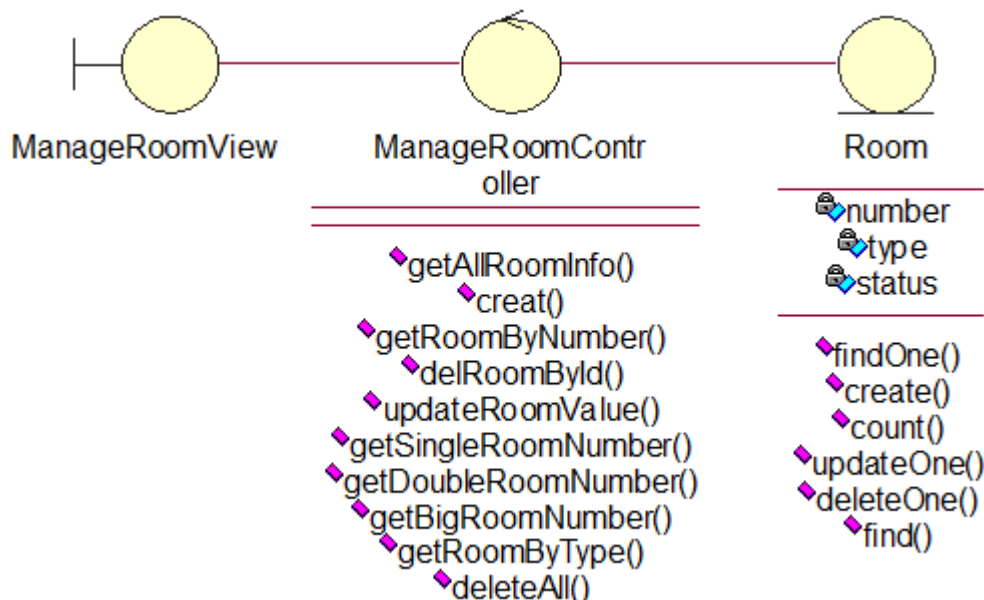


图 5-2-1 管理客房用例析取图

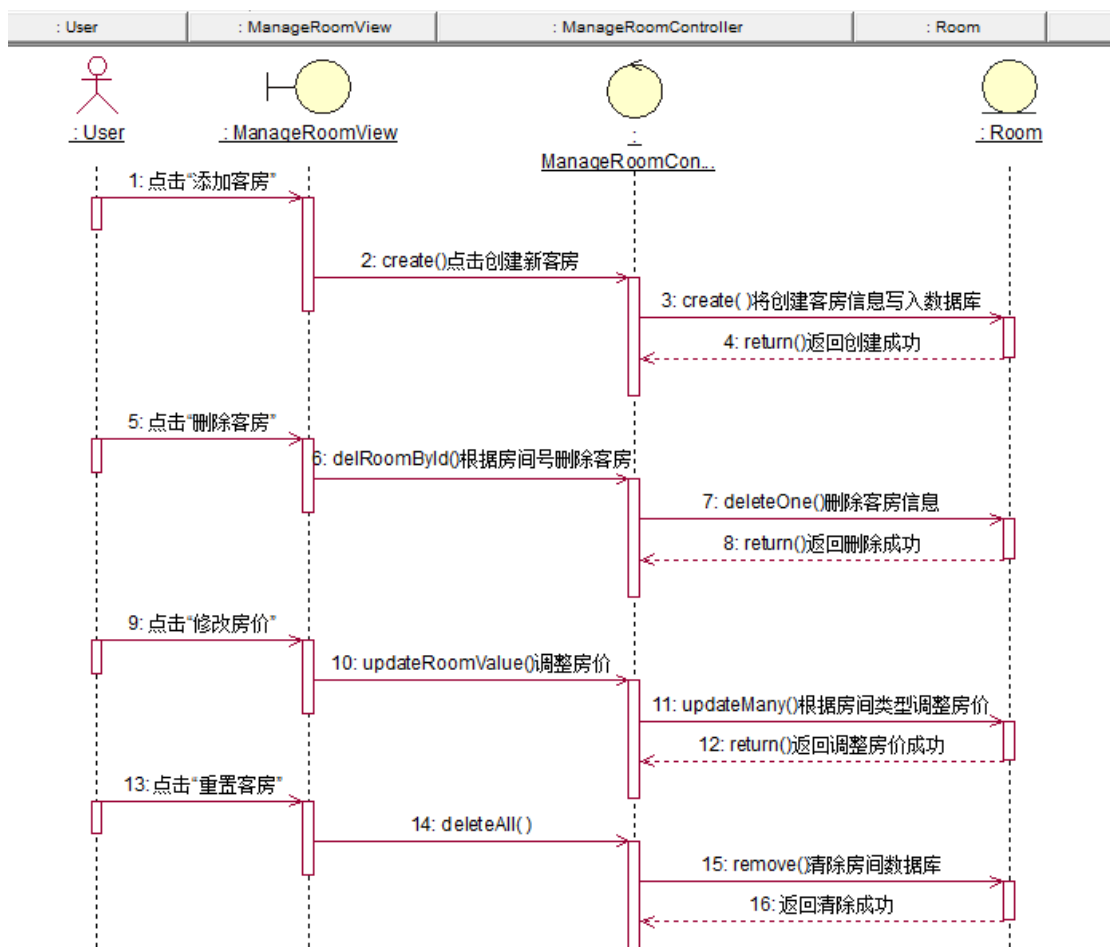


图 5-2-2 管理客房时序图

用户在系统功能主页面点击“管理客房”进行房间的相关管理操作。用户在“管理客房”的页面点击“添加客房”，页面跳转到创建页面，输入创建房间的相关信息，包括房间号、房间类型并管理员码。系统调用控制类的 `create()` 方法，然后调用实体类 `create()` 方法将创建的房间信息写入数据库并返回操作结果。如果填写无误返回“创建成功”信息并从创建页面回到“管理客房”页面，否则返回出错信息。其他操作以此类推。

< manageroom.ejs

该 ejs 文件对应管理客房主页的边界类 ManageRoomView

```

56  <!-- 左边导航栏end -->
57  <!-- 主要内容区start -->
58  <div class="main">
59    <div class="ui grid">
60      <div class="four wide column"></div>
61      <div class="eight wide column">
62        <div class="manage_room"><ul>
63          <li><a href="/manageroom/addroom"><h2>添加客房</h1></a></li>
64          <li><a href="/manageroom/updateroom"><h2>修改房价</h1></a></li>
65          <li><a href="/manageroom/deleteroom"><h2>删除客房</h1></a></li>
66          <li><a href="/manageroom/resetroom"><h2>重置客房</h1></a></li>
67        </ul></div>
68      </div>
69    </div>
70  </div>
71  <div class="blank"></div>

```

选择不同的管理选项点击链接实现页面的跳转

JS emptyRoomNumber.js

```

114 // 增加某个房间类型的数量 管理客房中增加客房时调用
115 addNumberByType: function addNumberByType(type) {
116   var today = new Date();
117   for(var i = 0; i < 30; i++) {
118     (function(dayoff) {
119       date = dateHelper.getDateAfterDays(today,dayoff)
120       module.exports.addNumberByDateAndType(date.year,date.month,date.day,type);
121     })(i);
122   }
123 },

```

这两个 js 对应管理客房主页的控制类 ManageRoomController

JS rooms.js

```

19
20 // 添加一个房间
21 create: function create (room) {
22   return Room.create(room).exec()
23 },

```

Room 对应管理客房用例的实体类 Room

管理客房用例的控制类 ManageRoomController 分 emptyRoomNumber.js 和 rooms.js 两个文件实现, emptyRoomNumber.js 实现对可用空房的操作, rooms.js 实现对房间的增删改查的操作, 从而使其他用例也复用到对房间进行操作的部分代码。

5.2.3 查询会员用例析取

根据第一章系统需求分析中的用例规约, 经分析可得该用例中的三种类。

- 边界类: SearchGuestView。SearchGuestView 为管理员或前台招待员查询会员信息。
- 控制类: SearchGuestController。SearchGuestController 控制类负责处理

查询会员的相关操作，即按照身份证号进行查询，如果有返回查询结果说明该宾客已是酒店会员。

- 实体类：Guest。Guest 顾客类存储了与住宿宾客有关的信息，包括姓名、身份证号、联系方式和积分。

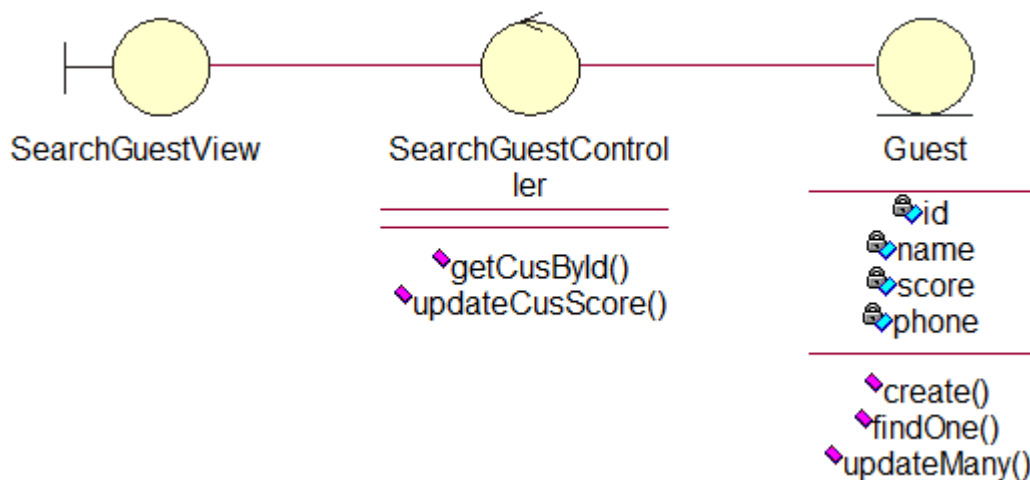


图 5-3-1 查询会员用例析取图

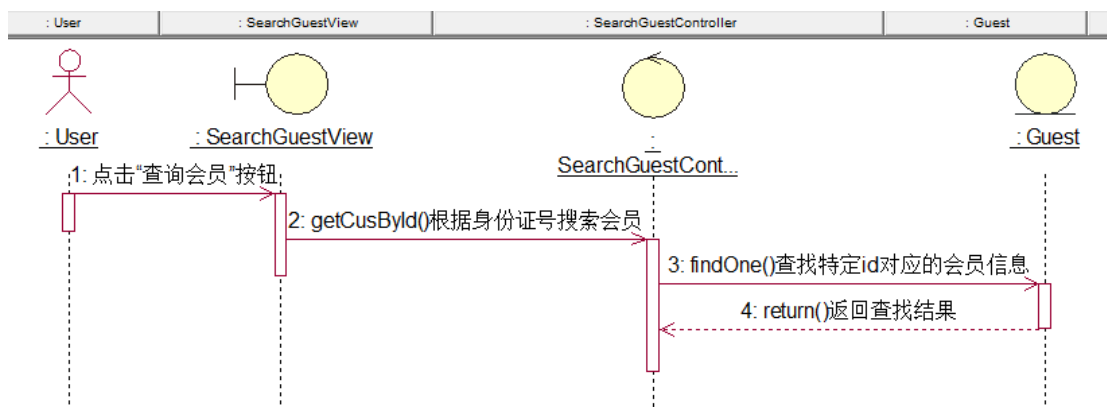


图 5-3-2 查询会员时序图

用户在系统功能主页面点击“查询会员”进行查询会员信息的操作。在该页面输入身份证号点击“查询会员”按钮，系统调用控制类的 `getCusById()` 方法，然后调用实体类 `findOne()` 方法查找特定 ID 对应的会员信息并返回操作结果。如果填写无误返回相关信息，否则返回出错信息。

```

searchcus.ejs
62
63
64
65
66
67
<form class="ui-form segment" method="post">
  <div class="field required">
    <label>身份证号</label>
    <input placeholder="身份证号" type="text" name="idcard" value="<%=customer.id%>" />
  </div>
  <input type="submit" class="ui button fluid" value="查询会员">

```



5.2.4 登记入住用例析取

根据第一章系统需求分析中的用例规约，经分析可得该用例中的三种类。

- 边界类：CheckinRoomView。CheckinRoomView 为管理员或前台招待员对入住的宾客进行信息登记和管理。
- 控制类：CheckinRoomController。CheckinRoomController 控制类负责处理登记入住宾客信息的相关操作，核对住房信息无误后输入会员号即可。
- 实体类：AvailableRoom、Room、Guest、Checkin 和 BookInfo。AvailableRoom 剩余房间类表示酒店可用房源，存储了当日起 30 天内可入住的房间信息，包括年月日基本时间信息，各种房间类型（大床房、单人房和双人房）可入住的数量。Room 房间类表示客房实体，存储酒店房间有关的信息，包括房间号、房间类型、房间价格和房间状态。Guest 用户类存储了与住宿宾客有关的信息，包括姓名、身份证号、联系方式和积分等。BookInfo 预订信息类存储了预订客房的相关信息，包括预订人的身份证号、联系方式、姓名，预订的房间类型，预订入住的起始时间和结束时间。Checkin 入住信息类存储了入住客房的相关信息，包括入住人的身份证号、姓名、联系方式，入住房间的房间号，入住时间和退房时间。

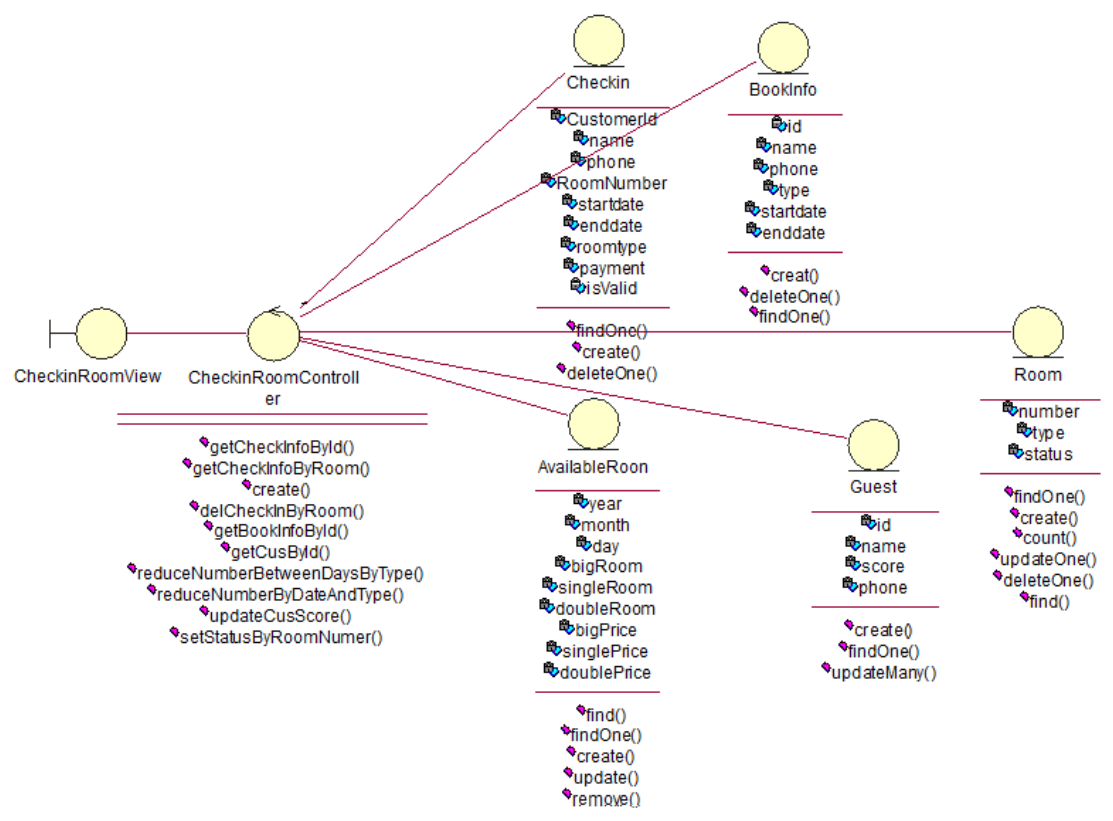


图 5-4-1 登记入住用例析取图

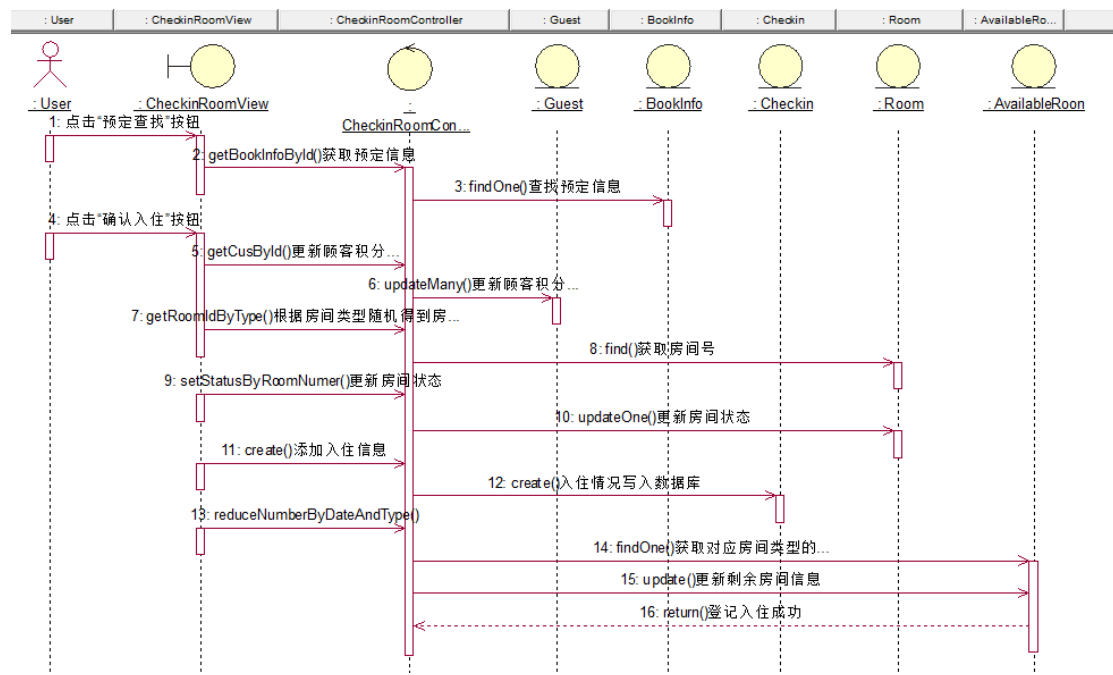


图 5-4-2-1 登记入住(有预订)时序图

用户在系统功能主页面点击“登记入住”进行入住手续办理，在“身份证号”一栏中输入顾客的身份证号，点击“预订查找”按钮。系统调用控制类的 `getBookInfoById()` 获取特定 `id` 的预订信息，然后调用 `BookInfo` 实体类的

findOne() 方法得到预订信息。点击“确认入住”按钮，系统调用控制类的 getRoomIdByType() 方法根据房间类型随机获取房间号，然后调用 Room 实体类的 find() 获取房间号。系统调用控制类的 setStatusByRoomNumber() 方法，根据房间号将该房间状态进行更新，然后调用 Room 实体类的 updateOne() 方法。系统调用控制类的 create() 方法添加入住信息，并在 Checkin 实体类中调用 create() 方法将入住信息写入数据库。系统调用控制类的 reduceNumberByDateAndType() 方法更新剩余房间信息，并调用实体类的 findOne() 方法和 update() 来实现。

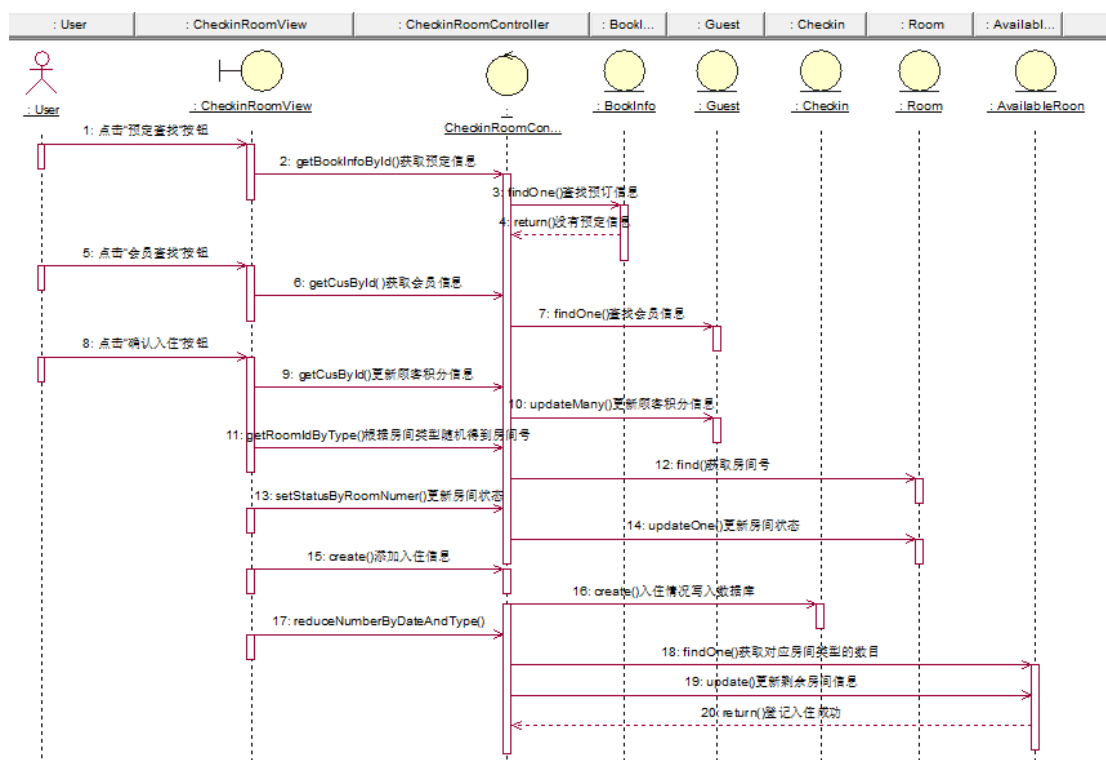


图 5-4-2-2 登记入住(无预订是会员)时序图

用户在系统功能主页面点击“登记入住”进行入住手续办理，在“身份证号”一栏中输入顾客的身份证号，点击“预订查找”按钮。系统调用控制类的 getBookInfoById() 获取特定 id 的预订信息，然后调用 BookInfo 实体类的 findOne() 方法返回“没有预订信息”。点击“会员查找”按钮系统调用控制类的 getCusById() 获取特定 id 的会员信息，并调用实体类的 findOne() 方法实现。点击“确认入住”按钮，系统调用控制类的 getRoomIdByType() 方法根据房间类型随机获取房间号，然后调用 Room 实体类的 find() 获取房间号。系统调用控制类的 setStatusByRoomNumber() 方法，根据房间号将该房间状态进行更新，然后

调用 Room 实体类的 updateOne() 方法。系统调用控制类的 create() 方法添加入住信息，并在 Checkin 实体类中调用 create() 方法将入住信息写入数据库。系统调用控制类的 reduceNumberByDateAndType() 方法更新剩余房间信息，并调用实体类的 findOne() 方法和 update() 来实现。

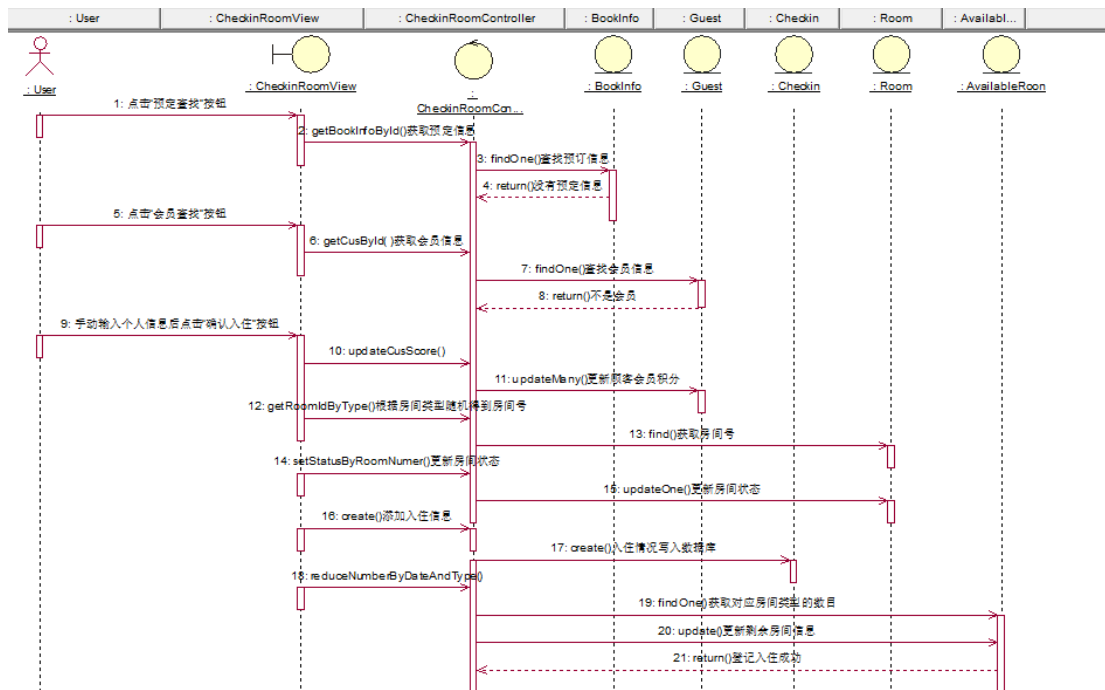


图 5-4-2-3 登记入住(无预订非会员)时序图

用户在系统功能主页面点击“登记入住”进行入住手续办理，在“身份证号”一栏中输入顾客的身份证号，点击“预订查找”按钮。系统调用控制类的 getBookInfoById() 获取特定 id 的预订信息，然后调用 BookInfo 实体类的 findOne() 方法返回“没有预订信息”。点击“会员查找”按钮系统调用控制类的 getCusById() 和实体类的 findOne() 方法未查找到会员信息。需要手动输入个人信息再点击“确认入住”按钮，系统调用控制类的 getCusById() 方法和 Guest 实体类的 updateMany() 方法更新顾客会员信息。接着，系统调用控制类的 getRoomIdByType() 方法根据房间类型随机获取房间号，然后调用 Room 实体类的 find() 获取房间号。系统调用控制类的 setStatusByRoomNumber() 方法，根据房间号将该房间状态进行更新，然后调用 Room 实体类的 updateOne() 方法。系统调用控制类的 create() 方法添加入住信息，并在 Checkin 实体类中调用 create() 方法将入住信息写入数据库。系统调用控制类的 reduceNumberByDateAndType()

方法更新剩余房间信息，并调用实体类的 `findOne()` 方法和 `update()` 来实现。



登记入住用例控制类 `CheckinRoomController` 包括 5 个 js 文件, 分别是 `bookinfo.js`、`customers.js`、`rooms.js`、`checkInfo.js` 和 `emptyRoomNumber.js`。每一文件对特定功能进行封装, 方便复用。



```
JS rooms.js x
49
50 // 用于登记入住管理:
51 // 根据房间类型随机获得空房的信息,可能返回有多个,类型: {number:xx, type:xx, value:xx, status:0}
52 getRoomIdByType: function getRoomIdByType (type) {
53     return Room.find({type:type, status: 0}).exec()
54 },
55
56 // 根据房间号码登记入住/退房(退房时customerId传入"0")
57 setStatusByRoomNumber: function setStatusByRoomNumber (number,customerId) {
58     return Room.updateOne({'number':Number(number)},{$set:{'status':customerId}}).exec()
59 },
```

```
JS checkInfo.js x
39
40 // 添加一个入住信息
41 create: function create (checkInfo) {
42     // 要计算钱
43     return CheckInfo.create(checkInfo).exec()
44 },
```

```
JS emptyRoomNumber.js x
64 // 减少某天某个房间类型的数量 预定时间调用/无预定入住时调用,参数type是string,其他是number
65 reduceNumberByDateAndType: function reduceNumberByDateAndType(year,month,day,type) {
66     EmptyRoomNumber.findOne({'year':year, 'month': month, 'day':day}).then(function(num) {
67         if (type == "大房" && num.bigRoom > 0) {
68             EmptyRoomNumber.update({'year':year, 'month': month, 'day':day},{ $set:{'bigRoom':num.bigRoom-1}}).exec()
69         } else if (type == "单人房" && num.singleRoom > 0) {
70             EmptyRoomNumber.update({'year':year, 'month': month, 'day':day},{ $set:{'singleRoom':num.singleRoom-1}}).
71         } else if (type == "双人房" && num.doubleRoom > 0) {
72             EmptyRoomNumber.update({'year':year, 'month': month, 'day':day},{ $set:{'doubleRoom':num.doubleRoom-1}}).
73         }
74     })
75 },
```

当点击页面“确认入住”按钮时调用 room.js 中的 getRoomIdByType() 函数,该函数通过填入的房间类型获取可用房间号;调用 checkInfo.js 中的 create() 函数,在入住数据库中增加一个入住信息;也调用了 emptyRoomNumber.js 中的 resuceNumberByDateAndType() 函数,根据入住房间类型更新某天某个房间类型的数量。

5.2.5 预订客房用例析取

根据第一章系统需求分析中的用例规约,经分析可得该用例中的三种类。

- 边界类: ReserveRoomView。ReserveRoomView 为管理员或前台招待员为宾客进行客房预订的操作。
- 控制类: ReserveRoomController。ReserveRoomController 控制类负责处理预订客房的相关操作,选中符号预订要求的客房并核对住房信息无误后,输入会员号完成预订操作。
- 实体类: AvailableRoom、BookInfo 和 Guest。AvailableRoom 剩余房间类表

示酒店可用房源，存储了当日起 30 天内可入住的房间信息，包括年月日基本时间信息，各种房间类型（大床房、单人房和双人房）可入住的数量。BookInfo 预订信息类存储了预订客房的相关信息，包括预订人的身份证号、联系方式、姓名，预订的房间类型，预订入住的起始时间和结束时间。Guest 用户类存储了与住宿宾客有关的信息，包括姓名、身份证号、联系方式和积分等。

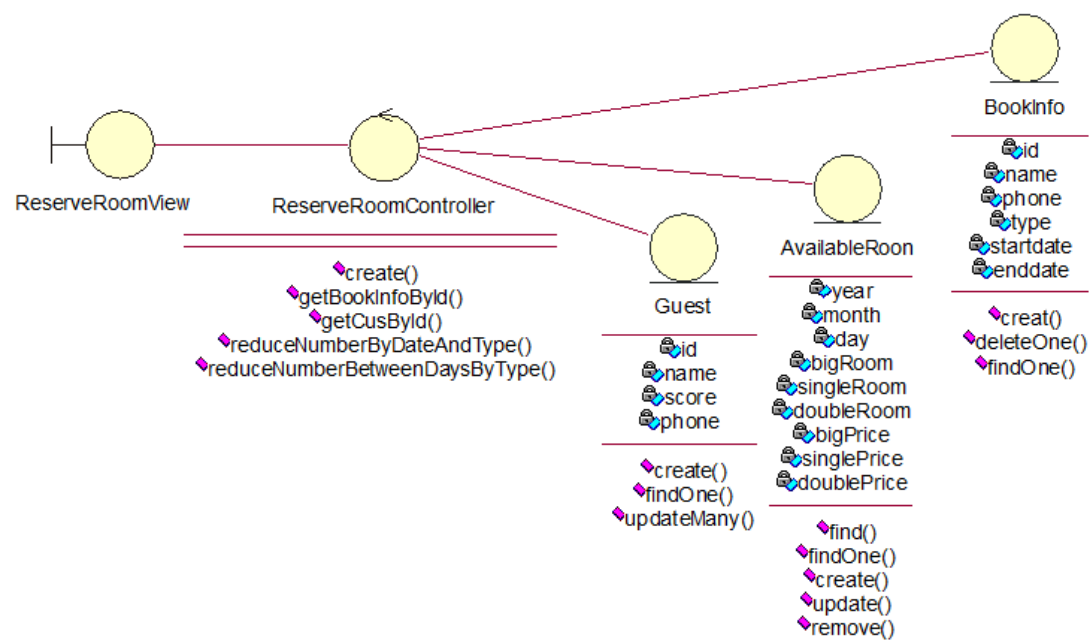


图 5-5-1 预订客房用例析取图

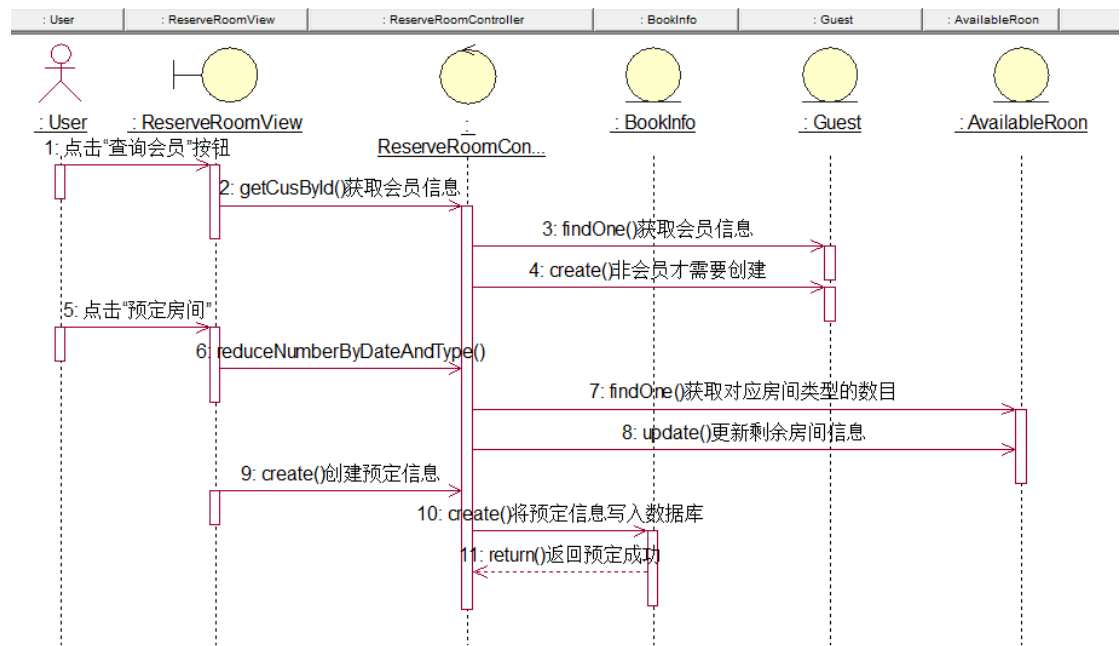


图 5-5-2 预订客房时序图

用户在系统功能主页面点击“预订客房”进行预订手续办理,在“身份证号”一栏中输入顾客的身份证号,点击“查询会员”按钮,系统调用控制类的 `getCusById()` 和 `Guest` 实体类的 `findOne()` 方法查找会员信息。如果未找到相关信息,需要手动输入个人信息并调用 `Guest` 实体类的 `create()` 方法。点击“预订房间”按钮,系统调用控制类的 `reduceNumberByDateAndType()` 方法更新剩余房间信息,并调用 `AvailableRoom` 实体类的 `findOne()` 方法和 `update()` 来实现。然后系统调用控制类和 `BookInfo` 实体类的 `create()` 方法,将预订信息写入数据库。

`<> bookroom.ejs` ×

bookroom.ejs 对应预定客房的边界类 `ReserveRoomView`

```

63 <form class="ui form segment" method="post" id = "myFrm">
64   <div class="field required">
65     <label>身份证号</label>
66     <input placeholder="身份证号" type="text" name="idcard" value=<%=customer.id%>>
67   </div>
68
69   <input type="submit" class="ui button fluid" value="查询会员">

```

`<> bookroom.ejs` ×

```

90 </div>
91 <div class="field required">
92   <label>退房时间</label>
93   <input placeholder="退房时间" type="text" name="endtime" value="">
94 </div>
95 <input type="submit" class="ui button fluid" value="预订房间">

```

`JS bookInfo.js` ×

预定客房的控制类 `ReserveRoomController` 包括 `bookInfo.js`、`customers.js` 和 `emptyRoomNumber.js`

```

21 // 添加一个预定信息
22 create: function create (bi) {
23   var idx;
24   bookinfo = {id:bi.id, name:bi.name,phone:bi.phone,type:bi.type,startdate:bi.startdate,enddate:bi.enddate,i
25   return Promise.resolve()
26   .then(function(value) {
27     return module.exports.getIndex(bookinfo.id)
28   }).then(function(idx) {
29     bookinfo.idx = Number(idx);
30   }).then(function() {
31     // return Promise.resolve(idx)
32     return BookInfo.create(bookinfo).exec()
33   })
34 }
35 },

```

当点击页面“预定房间”按钮时触发 `bookInfo.js` 文件的 `create()` 函数,该函数实现在预定数据库中增加一条记录。也调用了 `emptyRoomNumber.js` 中的 `resuceNumberByDateAndType()` 函数,根据预定房间类型更新某天某个房间类型的数量。

`JS emptyRoomNumber.js` ×

```

64 // 减少某天某个房间类型的数量 预定时间(开始时间、结束时间、参数、是否包含其他房
65 reduceNumberByDateAndType: function reduceNumberByDateAndType(year,month,day,type) {
66   EmptyRoomNumber.findOne({'year':year, 'month': month, 'day':day}).then(function(num) {
67     if (type == "大房" && num.bigRoom > 0) {
68       EmptyRoomNumber.update({'year':year, 'month': month, 'day':day},{ $set: {'bigRoom':num.bigRoom-1}}).exec()
69     } else if (type == "单人房" && num.singleRoom > 0) {
70       EmptyRoomNumber.update({'year':year, 'month': month, 'day':day},{ $set: {'singleRoom':num.singleRoom-1}}).
71     } else if (type == "双人房" && num.doubleRoom > 0) {
72       EmptyRoomNumber.update({'year':year, 'month': month, 'day':day},{ $set: {'doubleRoom':num.doubleRoom-1}}).
73     }
74   })
75 },

```



5.2.6 登记退房用例析取

根据第一章系统需求分析中的用例规约，经分析可得该用例中的三种类。

- 边界类：CheckoutRoomView。CheckoutRoomView 为管理员或前台招待员对退房的宾客进行信息登记和管理。
- 控制类：CheckoutRoomController。CheckoutRoomController 控制类负责处理进行退房结账的宾客信息的相关操作，核对住房信息和会员号无误后完成住宿费的支付即可。
- 实体类：AvailableRoom 和 Checkin。AvailableRoom 剩余房间类表示酒店可用房源，存储了当日起 30 天内可入住的房间信息，包括年月日基本时间信息，各种房间类型（大床房、单人房和双人房）可入住的数量。Checkin 入住信息类存储了入住客房的相关信息，包括入住人的身份证号、姓名、联系方式，入住房间的房间号，入住时间和退房时间。

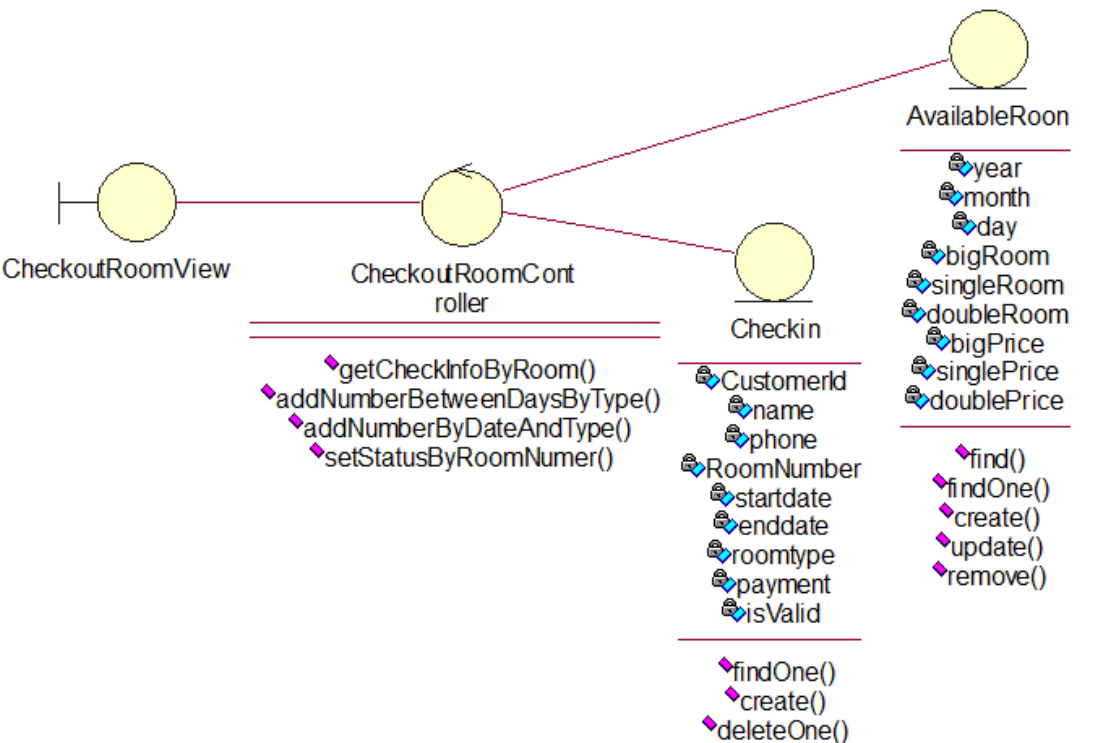


图 3-6-1 登记退房用例析取图

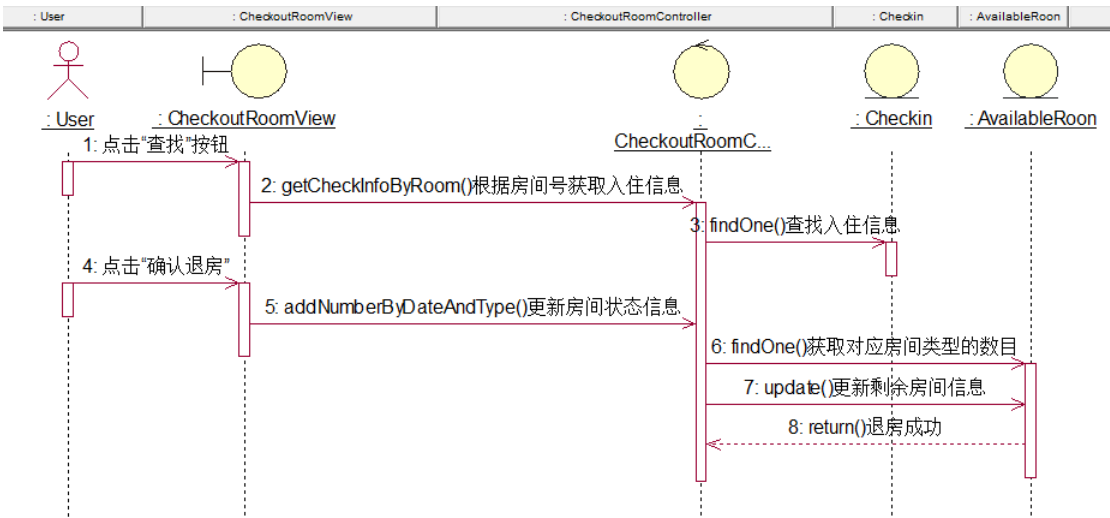


图 3-6-2 登记退房时序图

用户在系统功能主页面点击“登记退房”进行退房手续办理，在“房间号”一栏中输入需要退房的房间号，点击“查找”按钮，系统调用控制类的 `geCheckInfoByRoom()` 和 `Checkin` 实体类的 `findOne()` 方法查找入住信息。点击“确认退房”按钮，系统调用控制类的 `addNumberByDateAndType()` 方法更新剩余房间信息，并调用 `AvailableRoom` 实体类的 `findOne()` 方法和 `update()` 来实现。

```
checkout.ejs ✕ checkout.ejs 对应登记退房的边界类 CheckoutRoomView
92 </div>
93 <div class="field required">
94   <label>房费总计</label>
95   <input placeholder="房费总计" type="text" name="payment" value=<%=checkinfo.payment%>>
96 </div>
97 <input type="submit" class="ui button fluid" value="确认退房">
```

```
JS checkInfo.js ✕
12
13 // 通过房间号获取入住信息
14 getCheckInfoByRoom: function getCheckInfoByRoom (number) {
15   return CheckInfo
16     .findOne({ RoomNumber: number })
17     .addCreatedAt()
18     .exec()
19 },
```

```
JS emptyRoomNumber.js ✕
76
77 // 增加某天某个房间类型的数量 取消预定调用/退房调用 参数type是string,其他是number
78 addNumberByDateAndType(year,month,day,type) {
79   EmptyRoomNumber.findOne({'year':year, 'month': month, 'day':day}).then(function(num) {
80     if (type == "大房") {
81       EmptyRoomNumber.update({'year':year, 'month': month, 'day':day},{ $set:{ 'bigRoom':num.bigRoom+1}}).exec()
82     } else if (type == "单人房") {
83       EmptyRoomNumber.update({'year':year, 'month': month, 'day':day},{ $set:{ 'singleRoom':num.singleRoom+1}}).
84     } else if (type == "双人房") {
85       EmptyRoomNumber.update({'year':year, 'month': month, 'day':day},{ $set:{ 'doubleRoom':num.doubleRoom+1}}).
86     }
87   })
88 },
```

登记退房的控制类 CheckoutRoomController 包括 checkInfo.js 和 emptyRoomNumber.js 两个文件,在退房页面点击“查找”按钮时调用 checkInfo.js 中的 getCheckInfoByRoom() 函数,通过房间号获取入住信息,点击“确认退房”按钮是调用 emptyRoomNumber.js 中的 addNumberByDateAndType() 函数,增加某天某个房间类型的数量。

5.2.7 盘点结算用例析取

根据第一章系统需求分析中的用例规约,经分析可得该用例中的三种类。

- 边界类: BalanceView。BalanceView 为管理员删除过期的预订信息。
- 控制类: BalanceController。BalanceController 控制类负责处理盘点结算的相关操作,即清空过期的预订信息。
- 实体类: BookInfo。BookInfo 预订信息类存储了预订客房的相关信息,包括预订人的身份证号、联系方式、姓名,预订的房间类型,预订入住的起始时间和结束时间。

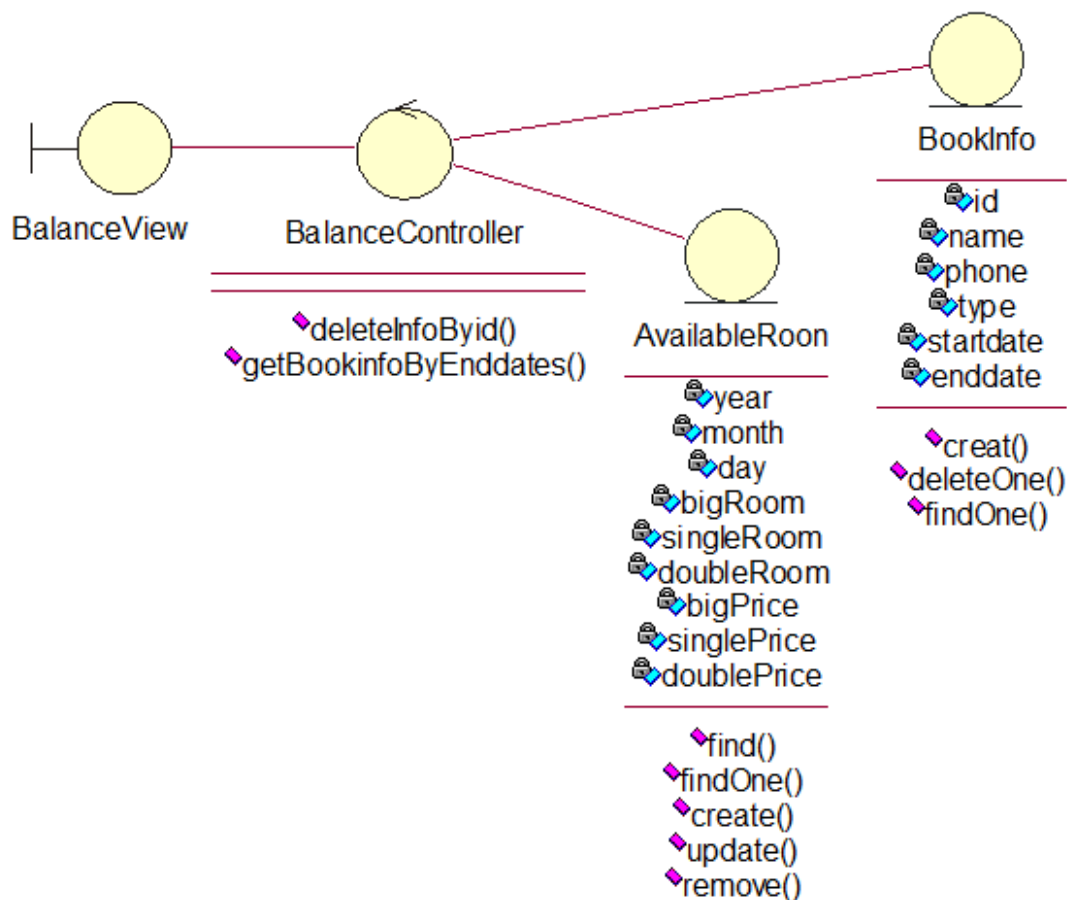


图 5-7-1 盘点结算用例析取图

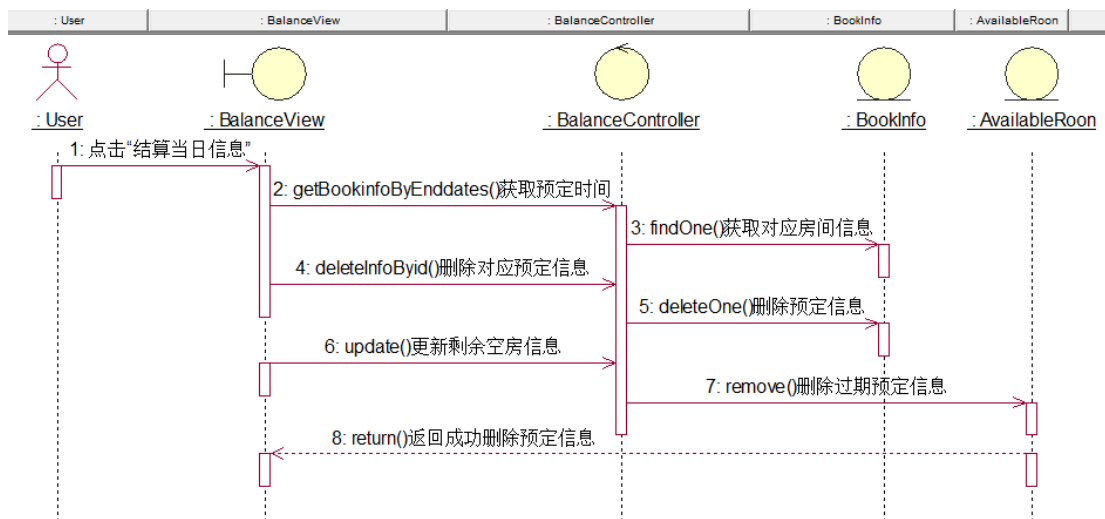
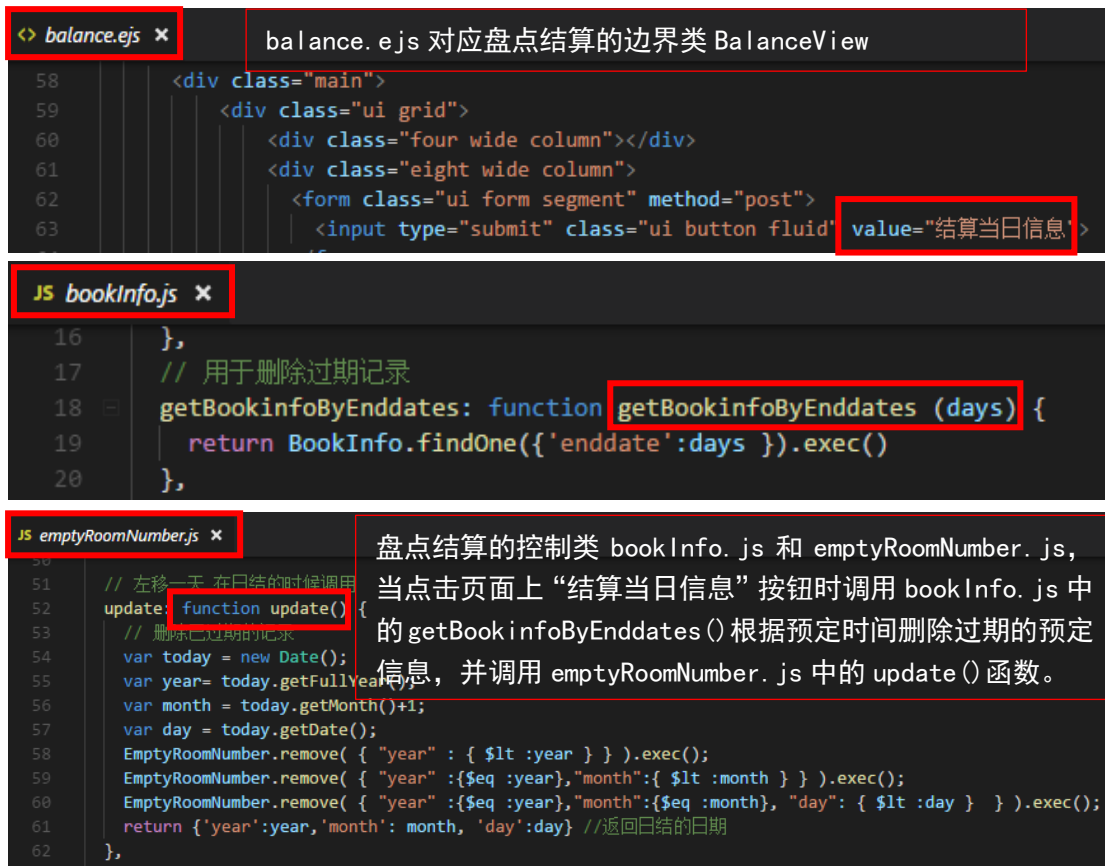


图 3-7-2 盘点结算时序图

用户在系统功能主页面点击“盘点结算”进行过期预订信息处理，点击“结算当日信息”按钮，系统调用控制类的 `getBookinfoByEnddates()` 和 `BookInfo` 实体类的 `findOne()` 方法获取过期的预订信息。然后调用控制类的 `deleteInfoByid()` 方法和 `BookInfo` 实体类的 `deleteOne()` 方法删除预订信息。

接着调用控制类的 `update()` 方法和 `AvailableRoom` 实体类的 `remove()` 方法更新剩余房间信息。



5.2.8 生成财务报表用例析取

根据第一章系统需求分析中的用例规约, 经分析可得该用例中的三种类。

- 边界类: `FinancialChartView`。`FinancialChartView` 为管理员获取酒店经营当天的盈利情况。
- 控制类: `FinancialChartController`。`FinancialChartController` 控制类负责处理生成财务报表的相关操作, 即当天住宿结账的房间信息和账单。
- 实体类: `Checkin`。`Checkin` 入住信息类存储了入住客房的相关信息, 包括入住人的身份证号、姓名、联系方式, 入住房间的房间号, 入住时间和退房时间。

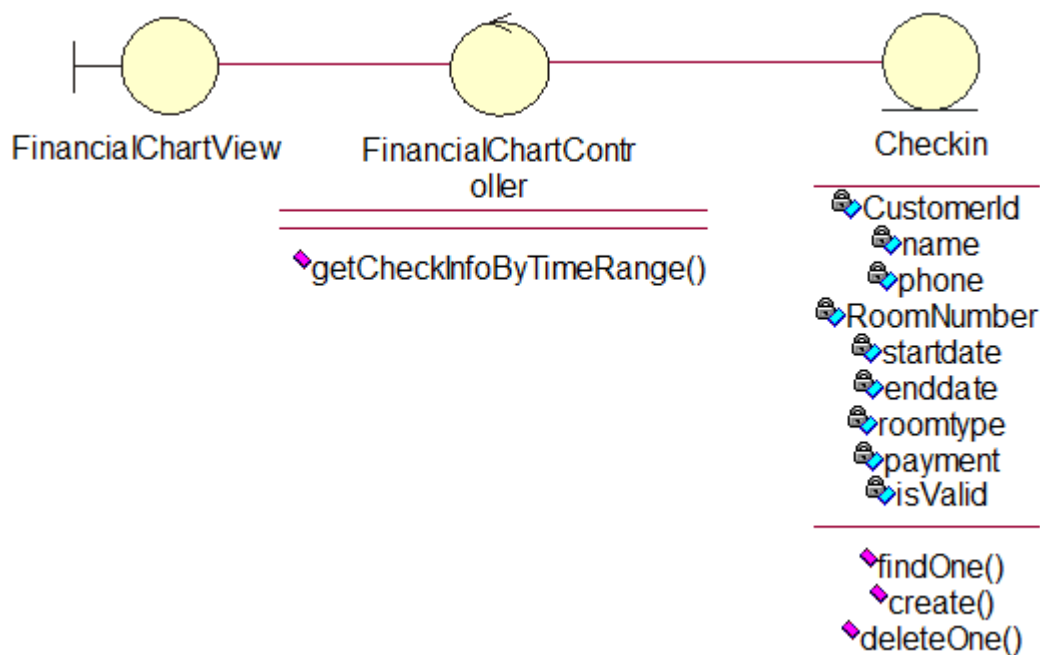


图 5-8-1 生成财务报表用例析取图



图 5-8-2 生成财务报表时序图

用户在系统功能主页面点击“生成报表”进行生成特定时间段盈利情况的报表。在该页面输入起始时间获取入住信息，在输入管理员码后点击“生成财务报表”按钮，系统调用控制类的 `getCheckInfoByTimeRange()` 和 `Checkin` 实体类的 `find()` 方法获取指定时间段的入住信息，再点击“下载 excel”即可。

```

<> finance.ejs x
finance.ejs 对应生成财务报表的边界类 FinancialChartView
78
79
80 <input type="submit" class="ui button fluid" value="生成财务报表"
81 </form>
82 </div>
83 </div>
84 </div>

```

当点击“生成财务报表”按钮是触发控制类的 `getCheckInfoByTimeRange()` 函数，通过起止时间获取入住信息

JS checkInfo.js x

生成财务报表的控制类 FinancialChartController 包括 checkInfo.js

```
32
33 //通过起止时间获取入住信息
34 getCheckInfoByTimeRange: function getCheckInfoByTimeRange(startdate,enddate) {
35     return CheckInfo
36         .find({ "startdate" : {$gte:startdate}, "enddate":{$lte:enddate}})
37         .exec()
38 },
```

5.3 分析机制

本节根据 1.4 补充规约得到上述边界类、控制类、数据类需要满足的非功能性需求，列出系统的分析机制表，如下表所示。

表 5-1 酒店住宿管理系统分析机制表

分析类 (analysis classes)	分析机制 (analysis mechanism)
Guest	持久性、安全性
BookInfo	持久性、安全性
User	持久性、安全性
Room	持久性、安全性
Check in	持久性、安全性
AvailableRoom	持久性、安全性

5.4 合并分析类

本节将析取出来的边界类、控制类、数据类进行合并整理，得到系统的合并类图。

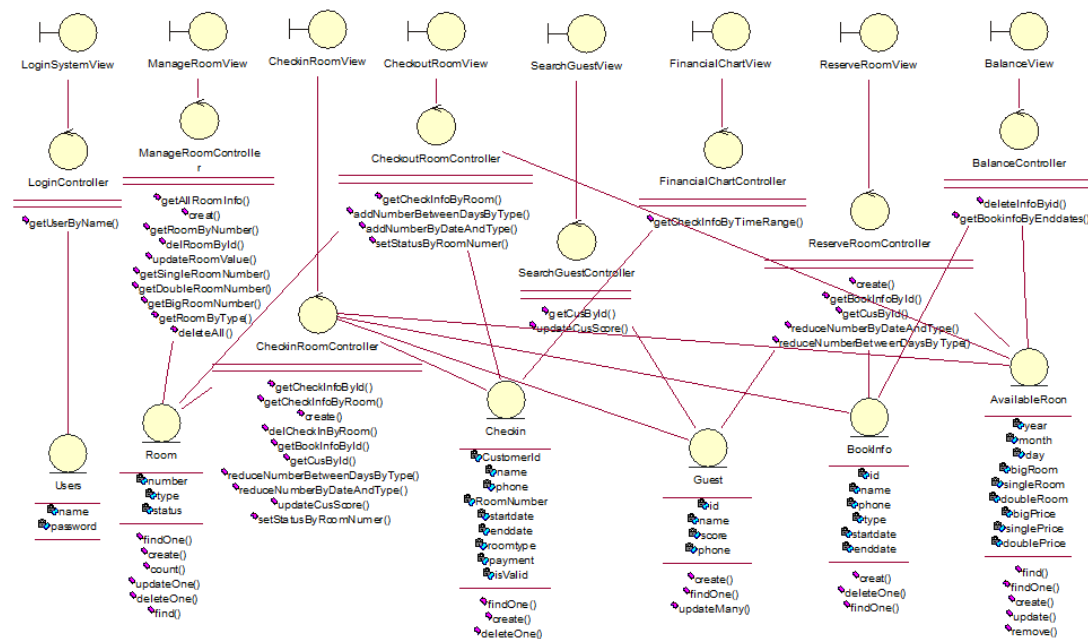


图 3-9 酒店住宿管理系统合并类图

六、子系统及其接口设计

6.1 确定设计类

本项目组对第三章的分析类进行了分析与检查，以确定其是否能成为设计类。经过分析发现，第三章中所有分析类均为单逻辑，不需要进行类的分解或合并，因此不做修改，设计类如第三章所示。

6.2 划分子系统

经过本项目组分析，本系统无需进行子系统设计及其接口设计。

七、部件设计

第五章类图中并未规定系统运行时功能，因此在系统运行时可能因为多线程多进程并发需求等产生资源抢占等问题。部件设计会分析系统运行时产生的一系列非功能性问题并给出解决方案。

7.1 分析并发需求

在一个系统中，并发性往往是至关重要的。因为如果有多个 CPU（Central Processing Unit）可供使用，则并行地执行任务可以提高系统性能，同样系统的并发性也可以用来响应外部的随机发生的事件，并发性也可以增强系统的控制能力。

系统的并发需求主要来源于以下几个方面：系统分布运行的程度、系统事件驱动的程度、关键算法的计算密度、运行环境支持的并发执行的程度。

常用的并发机制有以下几种：多进程，应用程序在多个 CPU 上并行执行；多任务：操作系统通过间断性地执行不同的任务来实现并发。基于应用程序的并发：应用软件在适当的时间转入到不同的分支中。

本系统在运行时会产生一些并发需求，比如

- (1) 住房人数超过酒店限制导致房源不足
- (2) 数据库能否及时更新问题

7.2 针对特定需求的设计方案

7.2.1 针对住房人数超过酒店限制导致房源不足的解决方案设计

在酒店经营高峰期可能出现供不应求的情况，入住需要的房间数可能会超过酒店所能提供的房间数，此时会产生资源冲突问题。为了解决这个问题，我们对有入住需求的宾客在登记入住前会进行酒店可用房源的查找。如果酒店有满足其入住需求的空房，则可以顺利登记入住。

7.2.2 针对数据库能否及时更新的解决方案设计

下面以其中一个函数为例进行解释：通过 `addCreatedAt()` 添加时间戳，指明操作进行的先后顺序。

```
// 通过客户id获取用户信息
getCusById: function getCusById (id) {
  return Customer
    .findOne({ id: id })
    .addCreatedAt()
    .exec()
},
```

八、测试用例设计

8.1 等价类划分

等价类划分是把所有可能的输入数据,即程序的输入域划分成若干部分(子集),然后从每一个子集中选取少数具有代表性的数据作为测试用例。等价类可划分两种不同情况,分别是有效等价类和无效等价类。

有效等价类是指对于程序的规格说明来说是合理的、有意义的输入数据构成的集合。利用有效等价类可检验程序是否实现了规格说明中所规定的功能和性能。

无效等价类是指对程序的规格说明是不合理的或无意义的输入数据所构成的集合。对于具体的问题,无效等价类至少应有一个,也可能有多个。

8.1.1 为登录系统设计等价类

表 8-1-1 测试登录系统的等价类

输入等价类	有效等价类	无效等价类
用户名	该用户名已注册	该用户名未注册
密码	与用户数据库中对应的密码一致	与用户数据库中对应的密码不一致

8.1.2 为管理客房设计等价类

表 8-1-2 测试管理客房的等价类

输入等价类	有效等价类	无效等价类
房间号	1、只能是数字 2、如果是“增加客房”,该房间号不能与已有房间号重复 3、如果是“删除客房”,该房间号必须存在	1、输入非数字 2、如果是“增加客房”,该房间号已存在 3、如果是“删除客房”,该房间号不存在
房间类型	输入“单人房/双人房/大房”	未输入“单人房/双人房/大房”
房间价格	只能是数字	输入非数字
开始时间	1、时间格式为“xxxxyyzz”,如20180601 2、输入时间不得早于当天	1、时间格式不是“xxxxyyzz” 2、输入时间早于当天
结束时间	时间格式为“xxxxyyzz”,如20180601	时间格式不是“xxxxyyzz”
管理员码	输入“forbidden”	输入非“forbidden”

8.1.3 为查询会员设计等价类

表 8-1-3 设计查询会员的等价类

输入等价类	有效等价类	无效等价类
身份证号	18 位数字	1、非数字 2、输入位数大于 18 位 3、输入位数小于 18 位

8.1.4 为预订客房设计等价类

表 8-1-4 设计预定客房的等价类

输入等价类	有效等价类	无效等价类
身份证号	18 位数字	1、非数字 2、输入位数大于 18 位 3、输入位数小于 18 位
联系方式	11 位数字	1、非数字 2、输入位数大于 18 位 3、输入位数小于 18 位
房间类型	输入“单人房/双人房/大房”	未输入“单人房/双人房/大房”
入住时间	1、时间格式为“xxxxyyzz”，如 20180601 2、输入时间不得早于当天	1、时间格式不为“xxxxyyzz” 2、输入时间早于入住时间
退房时间	1、时间格式为“xxxxyyzz”，如 20180601 2、输入时间不得早于入住时间	1、时间格式不为“xxxxyyzz” 2、输入时间早于入住时间

8.1.5 为登记入住设计等价类

由于登记入住和预订入住的等价类相同，在此不做赘述。

8.1.6 为登记退房设计等价类

表 8-1-5 设计登记退房的等价类

输入等价类	有效等价类	无效等价类
房间号	1、只能是数字 2、该房间号可在“登记入住”数据库中查找到	1、输入非数字 2、该房间号在“登记入住”数据库中无法查找到

8.1.7 为生成财务报表设计等价类

表 8-1-6 设计生成财务报表的等价类

输入等价类	有效等价类	无效等价类
入住时间	1、时间格式为“xxxxyyzz”， 如 20180601 2、入住时间不得晚于当天	1、时间格式非 “xxxxyyzz” 2、入住时间晚于当天
退房时间	1、时间格式为“xxxxyyzz”， 如 20180601 2、退房时间不得早于入住时间	1、时间格式非 “xxxxyyzz”， 2、退房时间早于入住时间
管理员码	输入为“forbidden”	输入非“forbidden”

8.2 边界值分析

边界值分析法就是对输入或输出的边界值进行测试的一种黑盒测试方法。通常边界值分析法是作为对等价类划分法的补充，这种情况下，其测试用例来自等价类的边界。

下面以“预定客房”为例分析边界值。

表 8-2 分析预定客房的边界值

输入项	边界值
身份证号	1、17 位数字 2、19 位数字
联系方式	1、10 位数字 2、12 位数字
入住时间	1、输入日期格式不符规定，如 2018-02-30 2、输入日期是登记入住当日
退房时间	1、输入日期格式不符规定，如 2018-02-30 2、输入日期等于入住时间的日期

8.3 测试用例

根据等价类划分和边界值分析两种方法，设计各用例测试样例如下所示。

8.3.1 登录系统测试用例

表 8-3-1 登陆系统测试用例表

测试项	条件	测试用例	预期结果
用户名	该用户名已注册	用户名：shey 密码：123456	成功
	该用户名未注册	用户名：she 密码：123456	失败
密码	与用户数据库中对应的密码一致	用户名：shey 密码：123456	成功

	与用户数据库中对应的密码不一致	用户名: shey 密码: 123311	失败
--	-----------------	-------------------------	----

8.3.2 管理客房测试用例

表 8-3-2 管理客房测试用例表

测试项	条件	测试用例	预期结果
房间号 (增加客房)	1、只能是 3 位数字 2、该房间号与已有房间号不重复	房间号: 123 房间类型: 单人房 管理员码: forbidden	成功
	输入非数字	房间号: sc2 房间类型: 单人房 管理员码: forbidden	失败
	输入字符串长度不为 3 位	房间号: 1234 房间类型: 单人房 管理员码: forbidden	失败
	该房间号与已有房间号重复	房间号: 001 房间类型: 单人房 管理员码: forbidden	失败
房间号 (删除客房)	1、只能是 3 位数字 2、该房间号已存在	房间号: 123 管理员码: forbidden	成功
	输入非数字	房间号: sc2 管理员码: forbidden	失败
	输入字符串长度不为 3 位	房间号: 1234 管理员码: forbidden	失败
	该房间号不存在	房间号: 124 管理员码: forbidden	失败
房间价格	只能是数字	起始时间: 20180615 结束时间: 20180701 房间类型: 单人房 房间价格: 300 管理员码: forbidden	成功
	输入非数字	起始时间: 20180615 结束时间: 20180701 房间类型: 单人房 房间价格: 30s 管理员码: forbidden	失败
房间类型 (以增加客房为例, 调	只能是“单人房/双人房/大房”中任一个	房间号: 123 房间类型: 单人房 管理员码: forbidden	成功
	输入不是“单人房/双人房/大房”中任一个	房间号: 123 房间类型: 单房	失败

整房价 同理)		管理员码: forbidden	
管理员 码 (以 删除客 房为 例, 其 他房间 操作同 理)	输入 “forbidden”	房间号: 123 管理员码: forbidden	成功
	输入字符串不为 “forbidden”	房间号: 123 管理员码: forbid	失败

8.3.3 查询会员测试用例

表 8-3-3 查询会员测试用例表

测试项	条件	测试用例	预期结果
身份证号	18 位字符	身份证号: 123456789098765432	成功
	字符串小于 18 位或大于 18 位	身份证号: 12345678909876543	失败

8.3.4 预定客房测试用例

表 8-3-4 预定客房测试用例表

测试项	条件	测试用例	预期结果
联系方式	11 位数字	姓名: 余滢 联系方式: 12345678909 房间类型: 单人房 入住时间: 20180612 退房时间: 20180614	成功
	11 位不全是数字	姓名: 余滢 联系方式: 123456ss909 房间类型: 单人房 入住时间: 20180612 退房时间: 20180614	失败
	数字串长度小于 11 或大于 11	姓名: 余滢 联系方式: 123456909 房间类型: 单人房 入住时间: 20180612 退房时间: 20180614	失败
身份证号	18 位字符	身份证号:	成功

		123456789098765432	
	字符串小于 18 位或大于 18 位	身份证号： 12345678909876543	失败
入住时间 (假设预定 当天日期 为 20180611)	1、时间格式 “xxxxyyzz” 2、时间格式中对月份限制在 1-12，即 yy 范围是 01-12 3、输入时间不早于预定当天	姓名：余滢 联系方式：12345678909 房间类型：单人房 入住时间：20180612 退房时间：20180614	成功
	时间格式非 “xxxxyyzz”	姓名：余滢 联系方式：12345678909 房间类型：单人房 入住时间：2018-06-12 退房时间：20180614	失败
	输入月份不处于 01-12	姓名：余滢 联系方式：12345678909 房间类型：单人房 入住时间：20181312 退房时间：20180614	失败
	输入时间早于入住/预定当天	姓名：余滢 联系方式：12345678909 房间类型：单人房 入住时间：20180610 退房时间：20180614	失败
退房时间	1、时间格式 “xxxxyyzz” 2、时间格式中对月份限制在 1-12，即 yy 范围是 01-12 3、输入时间不早于入住时间	姓名：余滢 联系方式：12345678909 房间类型：单人房 入住时间：20180612 退房时间：20180614	成功
	时间格式非 “xxxxyyzz”	姓名：余滢 联系方式：12345678909 房间类型：单人房 入住时间：20180612 退房时间：2018-06-14	失败
	输入月份不处于 01-12	姓名：余滢 联系方式：12345678909 房间类型：单人房 入住时间：20180612 退房时间：20181614	失败
	输入时间早于入住时间	姓名：余滢 联系方式：12345678909	失败

		房间类型：单人房 入住时间：20180612 退房时间：20180611	
房间类型 (以增加 客房为 例，调整 房价同 理)	只能是“单人房/双人房/ 大房”中任一个	姓名：余滢 联系方式：12345678909 房间类型：单人房 入住时间：20180612 退房时间：20180614	成功
	输入不是“单人房/双人 房/大房”中任一个	姓名：余滢 联系方式：12345678909 房间类型：单房 入住时间：20180612 退房时间：20180614	失败

8.3.5 登记入住测试用例

由于登记入住和与预订客房测试项与条件相同，在此不做赘述。

8.3.6 登记退房测试用例

表 8-3-5 登记退房测试用例表

测试项	条件	测试用例	预期结果
房间号	1、只能是 3 位数字 2、该房间号可在“登记入 住”数据库中查找到	房间号：123	成功
	输入非数字	房间号：sc2	失败
	输入字符串长度不为 3 位	房间号：1234	失败
	该房间号在“登记入住”数 据库中查找不到	房间号：001	失败

8.3.7 生成财务报表测试用例

表 8-3-6 生成财务报表测试用例表

测试项	条件	测试用例	预期结果
开始时间	1、时间格式“xxxxyyzz” 2、时间格式中对月份限制 在 1-12，即 yy 范围是 01-	开始时间：20180510 结束时间：20180530 管理员码：forbidden	成功

	12		
	时间格式不为“xxxxyyzz”	开始时间: 2018-05-10 结束时间: 20180530 管理员码: forbidden	失败
	输入月份不处于 01-12	开始时间: 20181510 结束时间: 20180530 管理员码: forbidden	失败
结束时间	1、时间格式“xxxxyyzz” 2、时间格式中对月份限制在 1-12, 即 yy 范围是 01-12 3、输入时间不早于开始时间	开始时间: 20180510 结束时间: 20180530 管理员码: forbidden	成功
	时间格式不为“xxxxyyzz”	开始时间: 20180510 结束时间: 2018-05-30 管理员码: forbidden	失败
	输入月份不处于 01-12	开始时间: 20180510 结束时间: 20181530 管理员码: forbidden	失败
	输入时间早于开始时间	开始时间: 20180510 结束时间: 20180501 管理员码: forbidden	失败
管理员码	输入“forbidden”	开始时间: 20180510 结束时间: 20180530 管理员码: forbidden	成功
	输入字符串不为“forbidden”	开始时间: 20180510 结束时间: 20180530 管理员码: forbid	失败