

# 分布式温控系统动态结构设计



姓名： 崔思颖 王宇杰 杨子旭 杨雨晨 戴骞 刘宝林

班级： 2018211305

组号： 305b 组

2021 年 5 月 23 日

## 目录

1. 文档介绍 .....	3
1.1. 文档目的 .....	3
1.2. 文档范围 .....	3
1.3. 读者对象 .....	3
1.4. 参考文档 .....	3
1.5. 分配情况 .....	3
2. 项目背景 .....	4
2.1. 项目需求来源 .....	4
2.2. 用户基本需求 .....	5
3. 动态结构设计 .....	5
3.1. 调度交互 .....	7
3.1.1. 使用空调 .....	7
3.2. 顾客 .....	13
3.2.1. 使用空调 .....	13
3.3. 管理员 .....	20
3.3.1. 运行空调 .....	20
3.3.2. 监视空调 .....	23
3.4. 前台 .....	25
3.4.1. 出详单 .....	25
3.4.2. 出账单 .....	27
3.5. 酒店经理 .....	29
3.5.1. 查看报表 .....	29

# 1. 文档介绍

## 1.1. 文档目的

本文档使用面向对象的方法在用例模型的基础上对分布式温控计费系统进行设计，实现了该系统的动态结构设计，为之后的静态结构设计以及软件开发奠定了基础。

## 1.2. 文档范围

围绕分布式温控系统展开，说明系统用途与需求，阐述规则与标准，全面介绍软件架构，分析该系统用例模型，尽可能详细的将每个用例逐一的进行解读、分析，构造对应的动态结构设计。

## 1.3. 读者对象

开发者，酒店管理者。

## 1.4. 参考文档

《系统解决方案》 305b 组第一次作业

《用户需求说明书》 305b 组第二次作业

《用例模型》 305b 组第三次作业

《软件工程模型与方法（第二版）》，肖丁、修佳鹏 编著 北京邮电大学出版社 2014

## 1.5. 分工情况

成员	学号	分工
崔思颖	2018211290	文档大纲的撰写；任务的分配与监督；调度部分的设计；文档的检查、修改、排版、导出
王宇杰	2018211266	文档介绍内容的撰写，管理员监视空调部分的设计
杨子旭	2018211298	项目背景内容的撰写，经理查看报表、打印报表部分的设计
杨雨晨	2018211278	前台查询详单账单、打印详单账单部分的设计
戴骞	2018211244	管理员运行空调部分的设计
刘宝林	2018211252	顾客使用空调部分的设计

## 2.项目背景

### 2.1. 项目需求来源

**1.用户调研：**通过用户初始给出的模糊的需求进行考察，确定基础的需求并实现，比如：入住的客户可以根据自身需求设定温度和风速的调节，同时房间内的控制面板显示所需支付的金额。客户退房时酒店须出具空调使用的账单及详单等功能。在软件开发或者软件运行的过程中还会随着用户需求的改变从而变更开发的细节。除此之外，我们会考虑对一些其他用户进行调研，考察一些具体的需求，修正开发过程。

**2.竞品分析：**通过了解其他类似的软件开发过程，判断其优劣程度，来寻找自己的突破口，对主体功能点，细节，每个页面，每个按钮，每个操作进行分析，对市场上的合格的产品进行融会贯通，了解市场的需求。

**3.数据分析与自我感知：**首先对用户初步的需求进行实现后，通过团队对结果的分析，判断哪里的数据不合理，哪里的显示需要改进。通过小组内成员

集思广益，符合自己心中预期。这个过程还需与用户进行双方的考量，决定最终的需求。

## 2.2. 用户基本需求

1. 入住的客户可以根据自身需求设定温度和风速的调节，同时房间内的控制面板显示所需支付的金额。
2. 客户退房时酒店须出具空调使用的账单及详单，详单的条目要全面。
3. 需要的情况下可以生成酒店及房间的空调使用统计报表。
4. 用户所使用的 UI 面板应该简洁明了，不杂乱。风速，温度，金额等条目均应清晰可见。
5. 用户信息的安全性要有保障，不能泄露。
6. 空调工作方式要满足用户需求，要让用户感受不到客房分机请求被调度。
7. 整个系统遇到错误要能够及时响应，不能耽误系统运行。
8. 管理员要能够监控所有房间的空调运行情况，及时排查异常。
9. 能够对系统进行定期的维护。
10. 具有高可靠性以及低的时延，不允许系统发生崩溃。

## 3.动态结构设计

类名说明：

层次	类名	说明
-	Customer	顾客，进行空调使用
-	Administrator	空调管理员
-	Receptionist	前台人员

	Manager	酒店经理
用户界面层	ControlPanel	房间空调控制面板界面，与顾客进行交互
用户界面层	ReceptionPanel	前台操作面板，与前台人员以及经理进行交互
控制器层	ReceptionController	在经理或前台人员使用系统的场景中，检查界面层请求的有效性，转发给应用逻辑层对象进行逻辑处理。
控制器层	UseController/ Scheduler	在顾客使用空调用例场景中，检查界面层请求的有效性，转发给应用逻辑层对象进行逻辑处理，并作为调度的对象。
控制器层	ManageController	在管理员运行空调用例场景中，检查界面层请求的有效性，转发给应用逻辑层对象进行逻辑处理。
业务逻辑层	ServiceList	服务对象组成的队列，负责创建服务对象的实例。
业务逻辑层	ServiceObj	服务对象，能够提供空调服务
业务逻辑层	MonitorList	监控对象组成的队列，负责创建监控对象的实例
业务逻辑层	MonitorObj	监控对象，能够监控空调运行
业务逻辑层	RequestList	请求对象组成的队列，负责创建请求对象的实例。
业务逻辑层	RequestObj	请求对象，记录客户使用空调的请求
业务逻辑层	TempUpList	服务对象组成的队列，负责创建提供室内温度变化处理的服务对象的实例。
业务逻辑层	TempUpObj	服务对象，能够提供调整室内温度变化处理
业务逻辑层	CenterServer	负责记录服务对象链接，接受，发送，处理报文。
业务逻辑层	dispatchServer	负责创建调度队列对象，维护房间对象组成的运行队列与等待队列。
业务逻辑层	RD	详单对象

业务逻辑层	RDFile	详单文件对象
业务逻辑层	Invoice	账单对象
业务逻辑层	InvoicFile	账单文件对象
业务逻辑层	Statistics	报表统计对象，经理查看报表时存储的结构
业务逻辑层	Report	报表打印对象，经理打印报表时输出的结构

## 3.1. 调度交互

### 3.1.1. 使用空调

■ **PowerOn(RoomId,CurrentTemp)->Return(Mode,CurrentTemp,TargetTemp,FanSpeed,RateFee,CurrentFee,TotalFee)**

参数：房间号，当前室温

返回值：工作模式，当前室温，当前目标温度，当前风速，费率，当前费用，总费用

#### □ 操作契约

系统事件	PowerOn(RoomId,CurrentTemp)
交叉引用	使用空调
前置条件	客户进入房间并插卡通电
后置条件	<ol style="list-style-type: none"> <li>1. 一个新的（概念类）空调对象实例被创建；</li> <li>2. 客户（概念类）与空调对象（概念类）建立关联；</li> <li>3. 一个新的（概念类）详单对象实例被创建；</li> <li>4. 一个新的（概念类）计时器对象被创建；</li> <li>5. 空调对象的状态、服务开始时间、工作模式、房间当前温度、目标温度、风速、费率、当前费用、总费用被赋值。</li> </ol>

#### □ 设计用例实现过程

1. 顾客 Customer 用控制面板 ControlPanel 发送开机 PowerOn 指令；

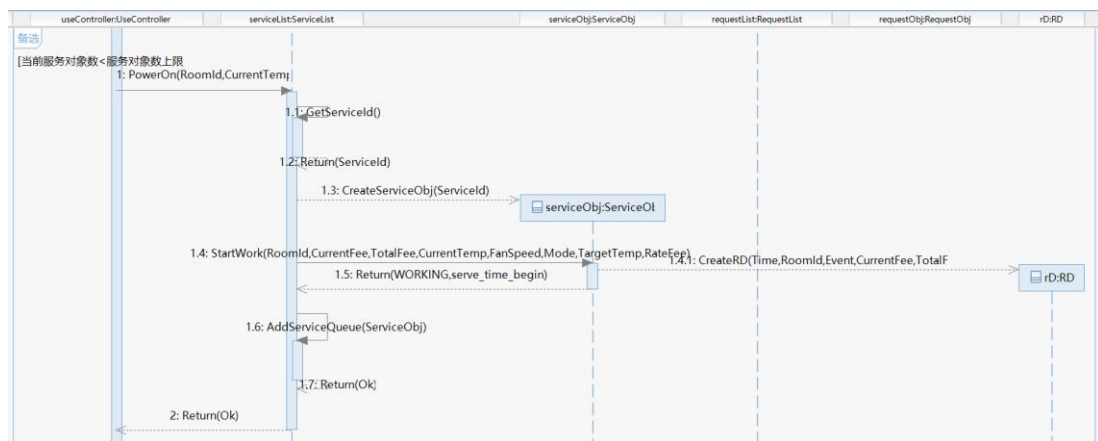
2. 选择 UseController 作为调度对象（即控制器对象），UseController 接收系统事件 PowerOn；
3. UseController 向请求中传入默认的服务参数（风速、模式、温度）；
4. 执行 UseController 的 GetNumService 方法，判断当前服务对象数是否达到上限。

#### 4.1. 当前服务对象数 < 服务对象数上限

4.1.1. UseController 将请求发送给 ServiceList，ServiceList 创建服务对象的实例 ServiceObj，为该 ServiceObj 分配 ID，并将该服务对象分配给这次请求；

4.1.2. ServiceObj 从请求中获取房间 ID、当前费用、总费用、房间当前温度、默认的服务参数（风速、模式、温度），并从 ServiceList 中获取相应费率；

4.1.3. ServiceList 调用 AddServiceQueue 方法将新创建的服务实例 ServiceObj 加入服务队列；



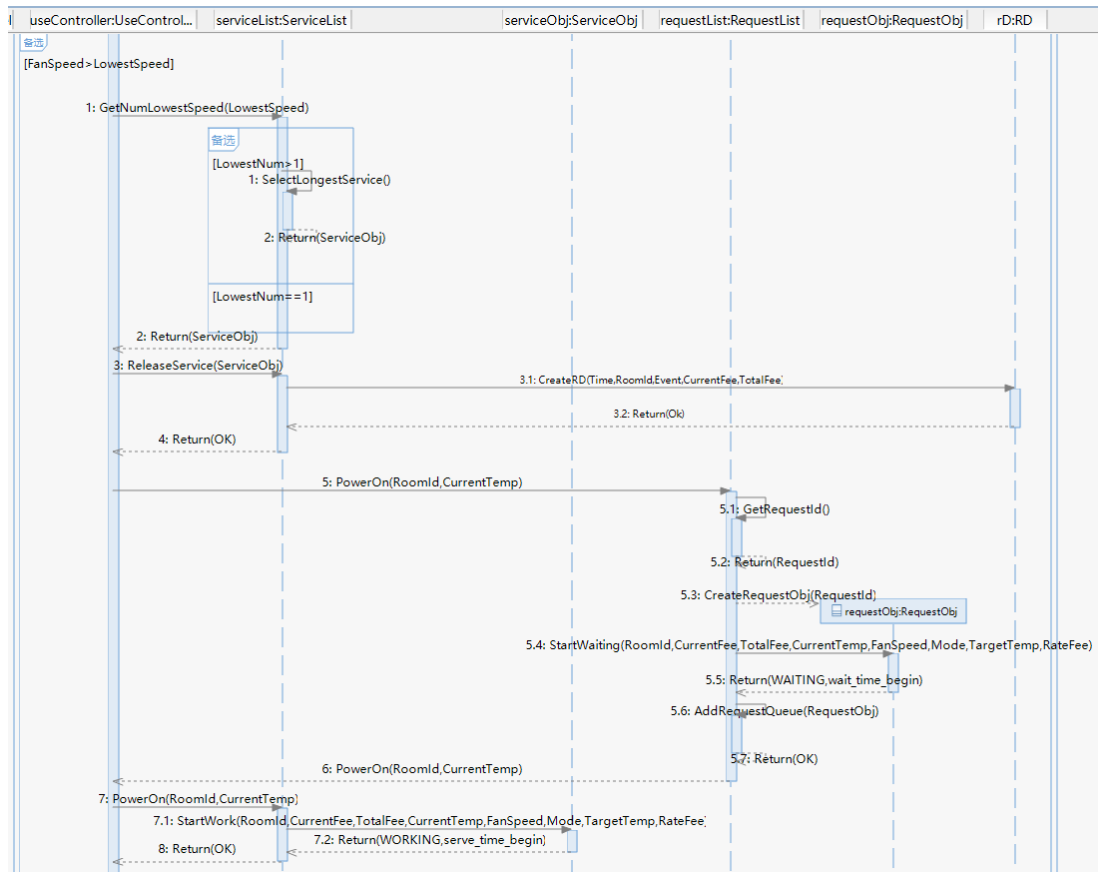
#### 4.2. 当前服务对象数 = 服务对象数上限



4.2.1. 调用 UseController 的 GetLowestSpeed 方法，得到当前服务队列的最低风速 LowestSpeed，由此判断当前服务请求能否获得资源，FanSpeed 为当前服务请求中的风速等级。

#### 4.2.1.1. FanSpeed > LowestSpeed

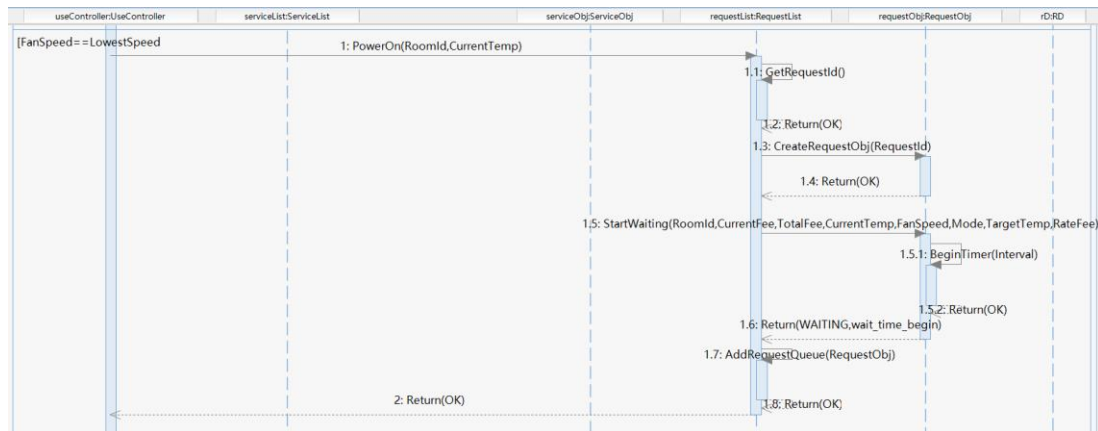
- 调用 ServiceList 的 GetNumLowestSpeed 方法，得到等于最低风速的正在被服务的请求的个数。
  - 若为 1，则调用 AddRequestQueue 方法，将该风速最低的房间请求放置于等待队列，相应的 ServiceObj 释放后分配给当前新的服务请求，并产生一条详单记录；
  - 若不为 1，则先调用 SelectLongestService 方法，找出当前服务时长最大的 ServiceObj，再调用 AddRequestQueue 方法，将对应的房间请求放置于等待队列，该 ServiceObj 释放后分配给当前新的服务请求，并产生一条详单记录。



#### 4.2.1.2. FanSpeed = LowestSpeed

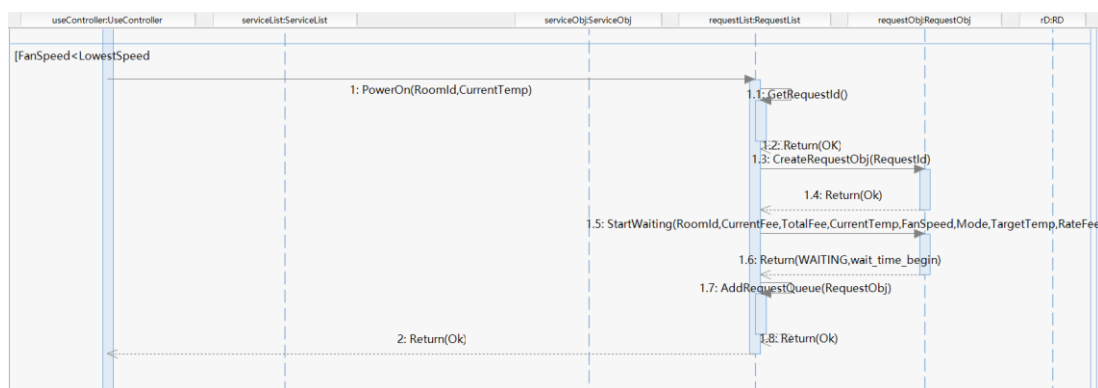
- UseController 将该请求发送给 RequestList, RequestList 创建等待对象 Request 的实例 RequestObj, 为该 RequestObj 分配 ID, 并将该等待对象分配给该请求;
- RequestList 用 AddRequest 方法将新创建的等待实例 RequestObj 加入等待队列;
- RequestObj 从请求中获取房间 ID、当前费用、总费用、房间当前温度、默认的服务参数 (风速、模式、温度);
- RequestObj 初始化时间片长度 Interval, 创建定时器对象 Timer 开始等待, 时长为一个 Interval, 当等待时间片耗尽时, Timer 会返回信号, 先调用 SelectLongestService 方法, 找出当前服务时长最大的 ServiceObj, 再调用 AddRe

quest 方法，将对应的房间请求放置于等待队列，该 ServiceObj 释放后分配给该服务请求。



#### 4.2.1.3. FanSpeed < LowestSpeed

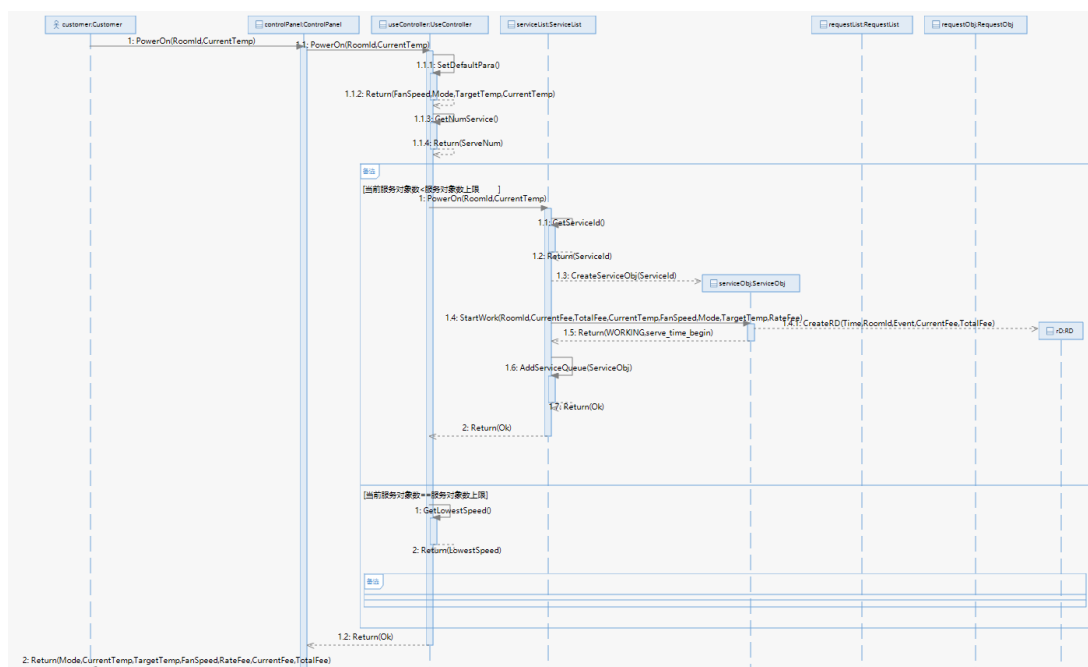
- UseController 将该请求发送给 RequestList, RequestList 创建等待对象的实例 RequestObj, 为该 RequestObj 分配 ID, 并将该等待对象分配给该请求;
- RequestObj 从请求中获取房间 ID、当前费用、总费用、房间当前温度、默认的服务参数 (风速、模式、温度), 开始等待, 直到有空闲的服务对象。
- RequestList 用 AddRequest 方法将新创建的等待实例 RequestObj 加入等待队列;



ServiceList 调用 ServiceObj 的 StartWorking 方法开始一次工作，ServiceObj 将服务开始时间 serve\_time\_begin 赋值为当前时间，开始提供服务，将服务状态置为“WORKING”，返回 ServiceObj 的服务状态 (WORKING) 和服务开始时间 (serve\_time\_begin)

5. 每开始一次工作，ServiceObj 会创建一条房间空调开启的详单记录。
6. RequestList 调用 RequestObj 的 StartWaiting 方法开始一次等待，RequestObj 将等待开始时间 wait\_time\_begin 赋值为当前时间，调用 state\_wait 方法开始等待，将等待状态置为“WAITING”，返回 RequestObj 的等待状态 (WAITING) 和等待开始时间 (wait\_time\_begin)
7. 当有 ServiceObj 的目标温度到达或关机，则调用 ChooseRequest 方法使该 ServiceObj 服务新的请求。

## □ Sequence Diagram



(备选片段详情见设计用例实现过程)

## 3.2. 顾客

### 3.2.1. 使用空调

■ **RequestState(RoomId)->Return(CurrentTemp,CurrentFee,TotalFee)**

参数：房间号

返回值：当前房间温度、当前费用、总费用

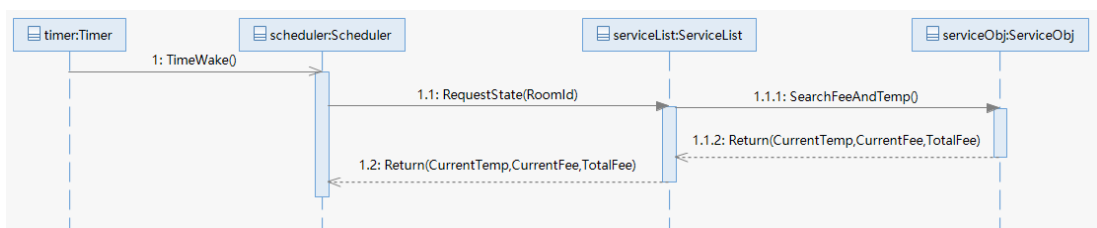
#### □ 操作契约

系统事件	ReqestState(RoomId)
交叉引用	使用空调
前置条件	空调正在运行
后置条件	1. 当前的空调对象实例的属性:房间当前温度、当前费用、总费用被修改;

#### □ 设计用例实现过程

1. Timer 定时发送查询信息给 scheduler
2. Scheduler 作为控制器对象接收系统事件 RequestState(RoomId);
3. ServiceList 根据房间号查询对应的服务对象 ID(RoomId);
4. ServiceList 向请求的房间对应的服务对象 ServiceObj 发送消息,查询房间的温度、当前费用、总费用;
5. 返回值为 ServiceObj 的服务相应属性值：房间温度，当前费用，总费用。

#### □ Sequence Diagram



## ■ RequestRelease(RoomId)->Return(state = Pause)

参数：房间号

返回值：服务状态（暂停状态）

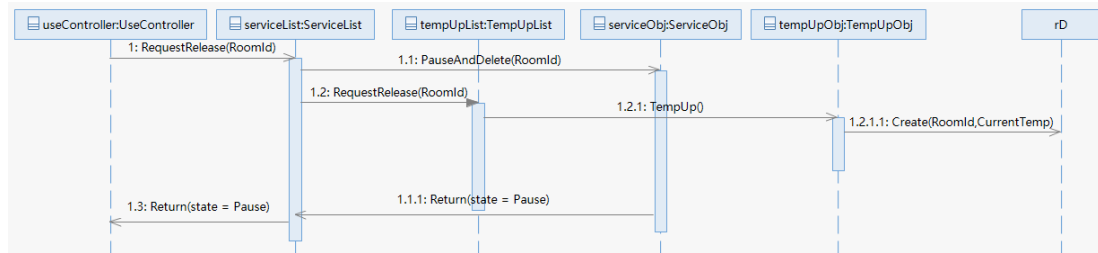
### □ 操作契约

系统事件	RequestRelease(RoomId)
交叉引用	使用空调
前置条件	空调正在运行
后置条件	1. 服务对象被删除; 2. 服务对象与客户对象之间的关联被删除;

### □ 设计用例实现过程

1. UseController 作为控制器对象接收系统事件 RequestRelease(RoomId) 并转发至 ServiceList
2. ServiceList 根据房间号查询对应的服务对象 ID(RoomId);
3. ServiceList 向 TempUpList 转发系统事件;
4. ServiceList 向请求的房间对应的服务对象 ServiceObj 发送消息，删除并释放该房间对应的服务对象 ServiceObj;
5. TempUpList 根据相应的房间号建立并初始化对象 TempUpObj，进行回温;
6. ServiceObj 创建一条该房间空调详单记录详单记录
7. 返回暂停状态

### □ Sequence Diagram



## ■ RequestTempUp()->Return(CurrentTemp)

参数： -

返回值：当前的温度

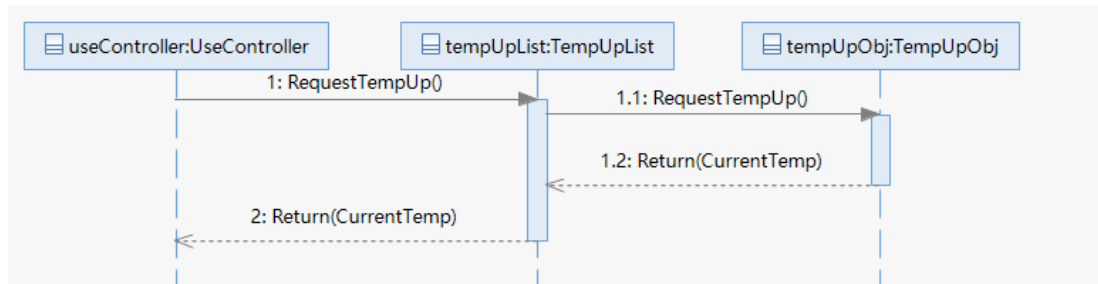
### □ 操作契约

系统事件	RequestTempUp()
交叉引用	使用空调
前置条件	空调暂停运行
后置条件	1. 服务对象被创建; 2. 服务对象与客户对象之间的关联被建立; 3. 服务对象的当前室温属性值被修改。

### □ 设计用例实现过程

1. UseController 作为控制器对象接收系统事件 RequestTempUp()并转发至 TempUpList
2. TempUpList 根据房间号查询对应的服务对象 ID(RoomId);
3. TempUpList 向请求的房间对应的服务对象 TempUpObj 发送消息，修改请求的房间对应的服务对象 TempUpObj 服务参数，调整其当前室内温度为最新设定的室内温度;
4. 返回值为室内温度 CurrentTemp

### □ Sequence Diagram



## ■ **PowerOff(RoomId)**

参数：房间号

返回值：-

### □ **操作契约**

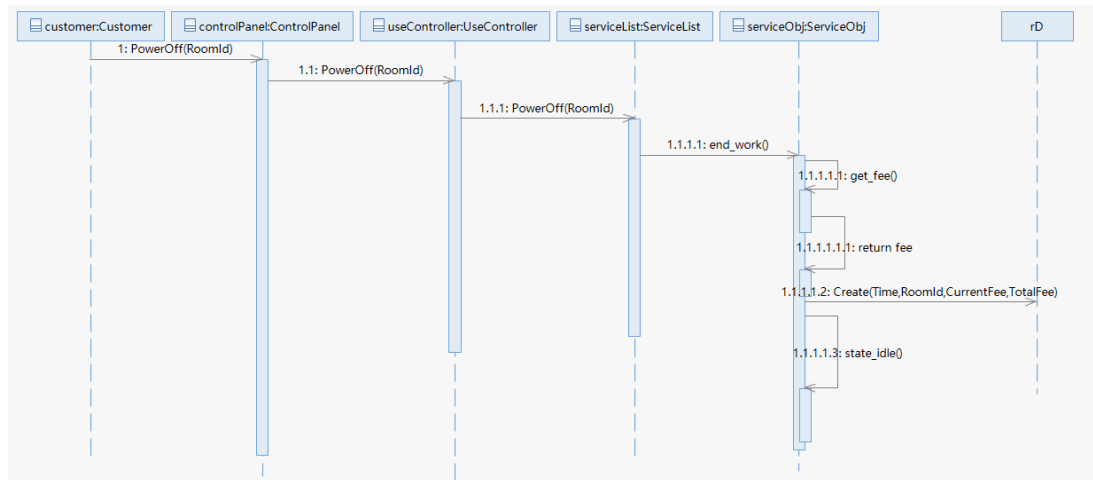
系统事件	PowerOff(RoomId)
交叉引用	使用空调
前置条件	空调正在运行
后置条件	1. 空调对象实例与详单建立关联; 2. 一条详单记录被写入到详单对象实例中(详单对象的各属性被赋值) 3. 空调对象实例被删除;

### □ **设计用例实现过程**

1. Customer 使用 ControlPanel 控制空调，发送关机信息 PowerOff(RoomId)
2. UseController 作为控制器对象接收系统事件 PowerOff(RoomId)并转发至 ServiceList
3. ServiceList 根据房间 ID 查询对应的服务对象的 ID
4. ServiceList 向请求的房间对应的服务对象 ServiceObj 发送消息，使停止工作：
  - 4.1 ServiceObj 获取本次服务的总费用
  - 4.2 ServiceObj 调用 state\_idle 方法，停止服务，将服务状态修改为空闲



## □ Sequence Diagram



## ■ ChangeFanSpeed(RoomId,FanSpeed) -> Return(RateFee)

参数：房间号，切换后的风速

返回值：切换后风速对应的费率

## □ 操作契约

系统事件	ChangeFanSpeed(RoomId,FanSpeed)
交叉引用	使用空调
前置条件	空调正在运行
后置条件	1. 当前空调对象实例的属性：风速、费率被修改； 2. 空调对象实例与详单（概念类）对象实例建立关联； 3. 一条详单记录被写入到详单（详单对象的各属性被赋值）

## □ 设计用例实现过程

1. Customer 在控制面板 ControlPanel 上点击按钮，切换风速档位为 FanSpeed，ControlPanel 向 UseController 发送 ChangeFanSpeed 请求。
2. 调度对象 UseController 调用 IsInServiceList 方法，判断当前服务请求的房间是否正在被服务。
  - 2.1. 请求的房间有服务对象

2.1.1. IsInService 返回服务对象 ID

2.1.2. UseController 向 ServiceList 发送 ChangeFanSpeed 请求, ServiceList 更改相应房间服务对象 ServiceObj 的服务参数, 风速改为 FanSpeed, 并向 ServiceObj 传入该风速对应的费率。

2.1.3. ServiceObj 创建一条该房间空调风速变化的详单记录。

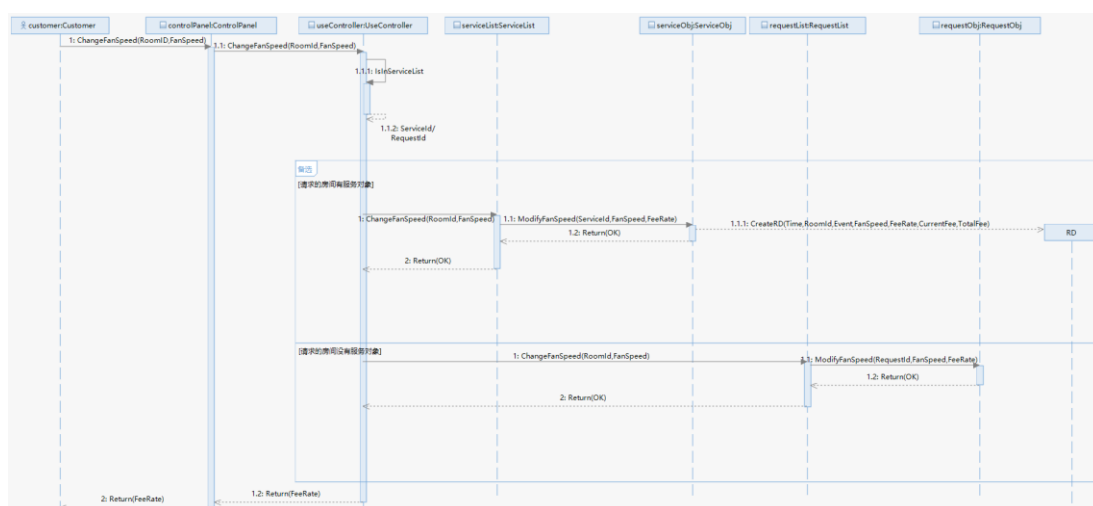
2.2. 请求的房间没有服务对象

2.2.1. IsInService 返回请求对象 ID

2.2.2. UseController 向 RequestList 发送 ChangeFanSpeed 请求, RequestList 更改相应房间服务对象 RequestObj 的服务参数, 风速改为 FanSpeed, 并向 RequestObj 传入该风速对应的费率。

3. ChangeFanSpeed 返回后, 系统将会执行一次调度算法 (PowerOn 设计用例实现中的步骤 4~8)

## □ Sequence Diagram



■ **ChangeTargetTemp(RoomId,TargetTemp)->Return(CurrentTemp)**

参数：房间号，目标温度

返回值：当前房间温度

#### □ 操作契约

系统事件	ChangeTargetTemp(RoomId,TargetTemp)
交叉引用	使用空调
前置条件	空调正在运行
后置条件	1. 当前空调对象实例的属性——目标温度被修改

#### □ 设计用例实现过程

1. Customer 在控制面板 ControlPanel 上点击按钮，切换温度档位为 TargetTemp，ControlPanel 向 UseController 发送 ChangeTargetTemp 请求。

2. 调度对象 UseController 调用 IsInServiceList 方法，判断当前服务请求的房间是否正在被服务。

##### 2.1.请求的房间有服务对象

2.1.1. IsInService 返回服务对象 ID

2.1.2. UseController 向 ServiceList 发送 ChangeTargetTemp 请求，ServiceList 更改相应房间服务对象 ServiceObj 的服务参数，修改其目标温度为 TargetTemp。

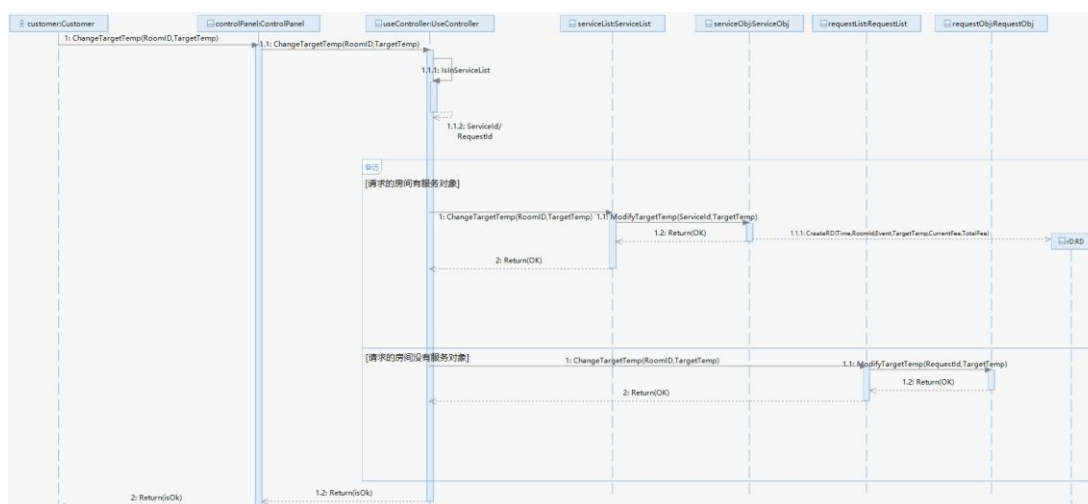
2.1.3. ServiceList 创建一条该房间空调目标温度变化的详单记录。

##### 2.2.请求的房间没有服务对象

2.2.1. IsInService 返回请求对象 ID

2.2.2. UseController 向 RequestList 发送 ChangeTargetTemp 请求，RequestList 更改相应房间服务对象 RequestObj 的服务参数，修改其目标温度为 TargetTemp。

### □ Sequence Diagram



## 3.3. 管理员

### 3.3.1. 运行空调

#### ■ PowerOn()->return(state)

参数： -

返回值：执行状态

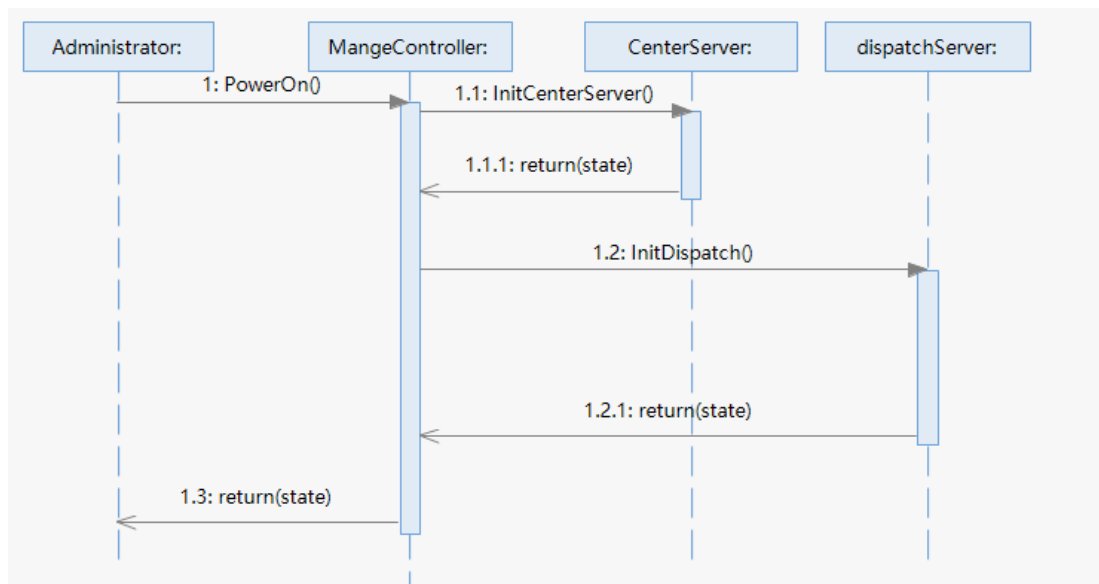
### □ 操作契约

系统事件	PowerOn()
交叉引用	打开中央空调
前置条件	中央空调未打开
后置条件	1. 管理员控制类对象被创建 2. 中央服务类对象被创建 3. 调度类对象被创建

### □ 设计用例实现过程

- 1.管理员打开空调 (PowerOn)
2. (MangeController) 管理员控制对象接受信号创建并初始化 (CenterServ  
er) 中央服务对象
3. (MangeController) 管理员控制对象创建并初始化 (dispatchServer) 调  
度服务对象
- 4.PowerOn 完成

#### □ Sequence Diagram



- **setPara(Mode,Temp\_highLimit,Temp\_lowLimit,default\_Tar  
getTemp,FeeRate\_H,FeeRate\_M,FeeRate\_L, MaxRunnin  
g) ->return(state)**

参数：模式，最高温度，最低温度，默认温度，最高费率，费率，最低  
费率，最大调度数

返回值：执行状态

#### □ 操作契约

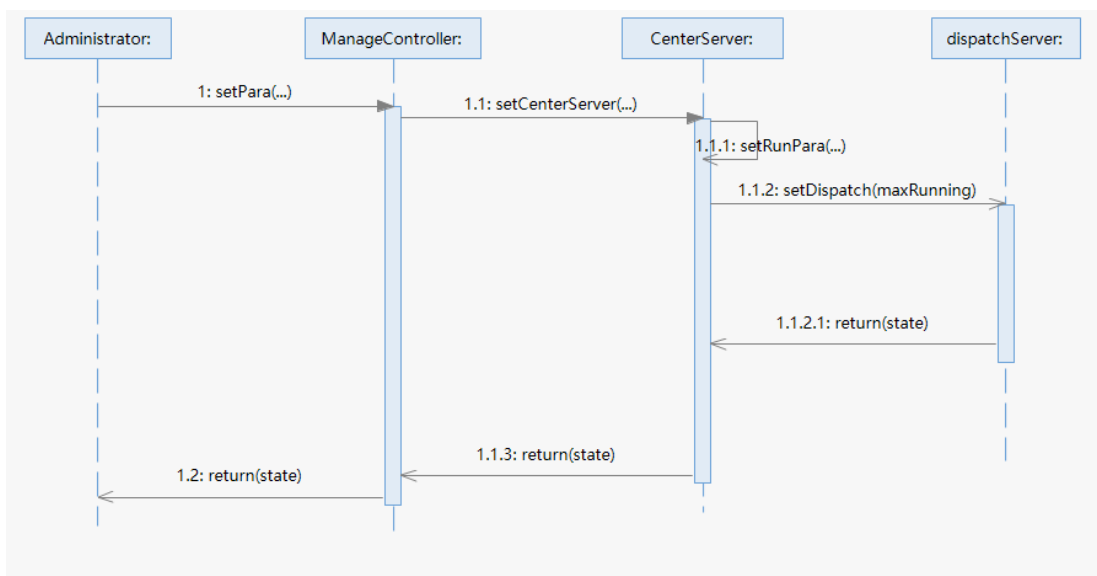
系统事件	setPara(Mode,Temp_highLimit,Temp_lowLimit,default_Targ
------	--

	etTemp, FeeRate_H, FeeRate_M, FeeRate_L, MaxRunning)
交叉引用	修改中央空调运行参数
前置条件	中央空调已打开
后置条件	1. 中央服务对象参数被修改 2. 调度对象参数被修改

#### □ 设计用例实现过程

1. 管理员设定空调各项运行参数
2. (MangeController) 管理员控制对象修改中央服务对象运行参数
3. (CenterServer) 中央服务对象修改调度服务对象运行参数
4. setPara 完成

#### □ Sequence Diagram



#### ■ StartUp()->return(state)

参数： -

返回值： 执行状态

#### □ 操作契约

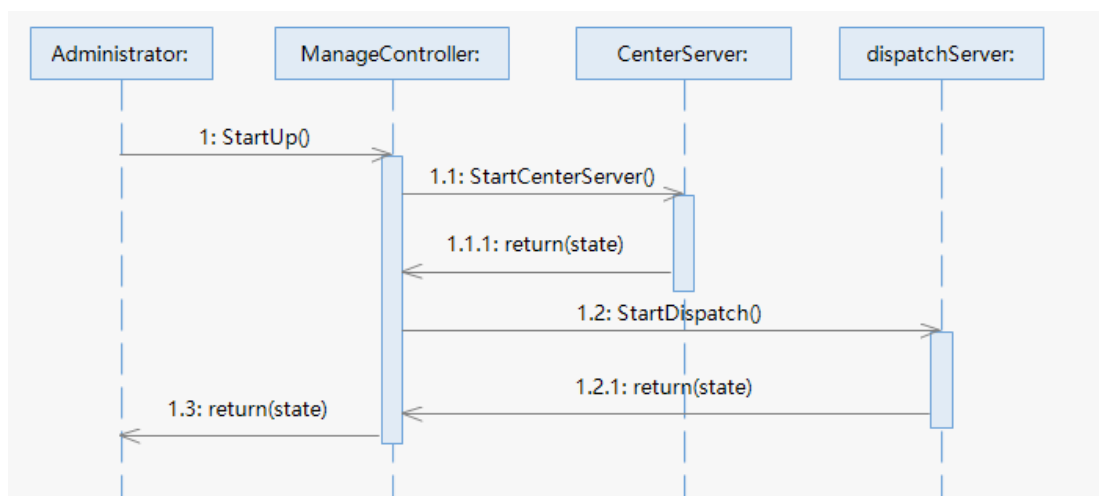
系统事件	StartUp()
交叉引用	启动中央空调

前置条件	中央空调已打开
后置条件	1. 中央服务对象与管理员控制对象建立关联 2. 管理员控制对象与调度对象建立关联

#### □ 设计用例实现过程

- 1.管理员启动中央空调
2. (ManageController) 管理员控制对象接收信号，启动 (CenterServer) 中央服务
3. (ManageController) 管理员控制对象启动 (dispatchServer) 调度服务
- 4.Startup 完成

#### Sequence Diagram



### 3.3.2. 监视空调

- CheckRoomState(list\_Room)->Return(Mode,Current\_temp,Target\_temp,Fan,Duration,Fee,Total\_fee,state)

参数：房间链表

返回值：模式，当前温度，目标温度，风速，服务时长，当前费用，总费用，房间服务状态

#### □ 操作契约

系统事件	CheckRoomState(list_Room)
交叉引用	监视空调
前置条件	空调对象开始运行
后置条件	1. 监控对象（概念类）被创建 2. 监控对象与服务对象（空调实例对象）建立关联 3. 监控对象的各属性被赋值：工作模式、当前室温、目标温度、风速、服务时长、当前费用、总费用、房间服务状态

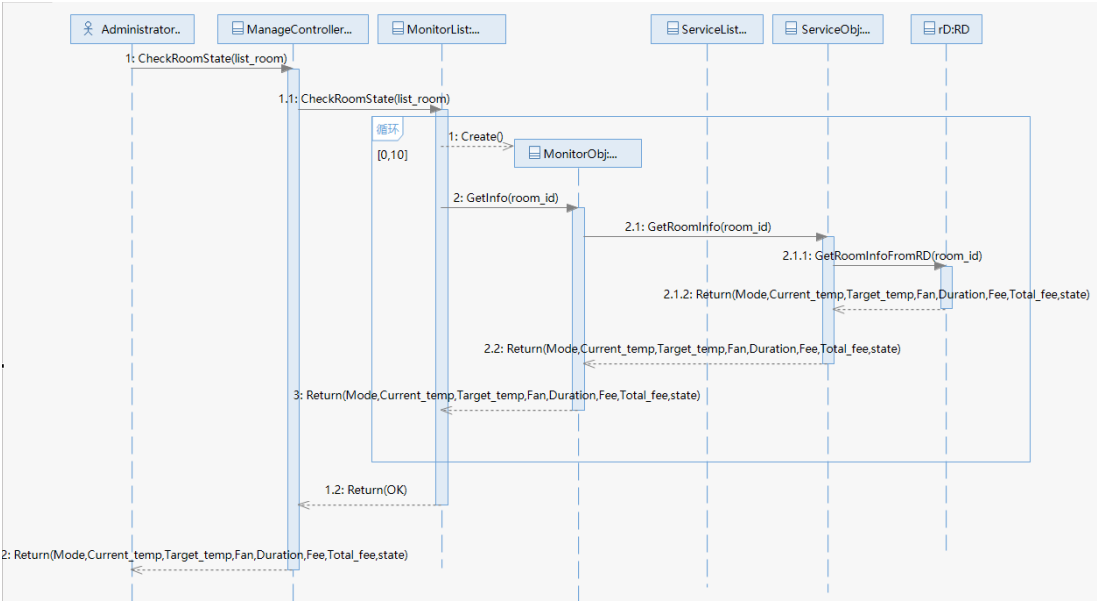
#### □ 设计用例实现过程

1. ManagerController 接收系统事件 CheckRoomState;
2. ManagerController 向 MonitorList（即业务逻辑层）发送 CheckRoomState 请求;
3. MonitorList 接收请求，循环创建 MonitorObj（监控对象），并发送 GetInfo(room\_id)请求;
4. MonitorObj 接受请求，与 ServiceObj（服务对象）建立关联并发送请求 GetRoomInfo(room\_id);
5. ServiceObj 收到请求向 RD（详单对象）发送 GetRoomInfoFromRD(room\_id)请求;
6. RD 接受请求返回数据;
7. ServiceObj 接收数据，返回数据;
8. MonitorObj 接收数据，添加到队列中去;
9. MonitorList 完成 CheckRoomState 请求，返回 ManageController;



10. CheckRoomState 完成。

Sequence Diagram



3.4. 前台

3.4.1. 出详单

CreateRD(RoomId,date\_in,date\_out)

操作契约

系统事件	CreateRD(RoomId,date_in,date_out)
交叉引用	出详单
前置条件	该房间对象已经被创建
后置条件	1.详单对象（概念类）被创建 2.详单对象与房间对象建立关联 3.详单对象（概念类）的属性被赋值：房间 id，入住时间，时刻，事件，本次费用，风速，请求时长

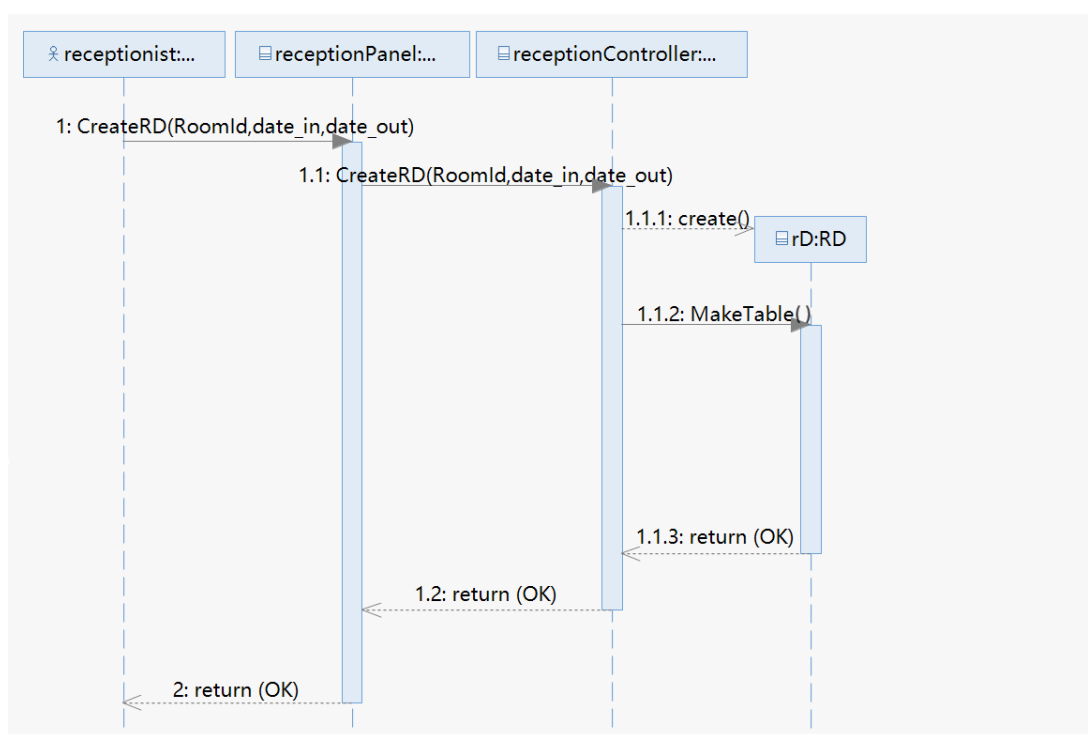
设计用例实现过程

1.前台人员点击操作前台面板（ReceptionPanel）

2.前台面板（ReceptionPanel）向前台控制器（ReceptionController）发送  
创建详单请求

3.前台控制器（ReceptionController）创建详单（RD）

#### □ Sequence Diagram



#### ■ PrintRD(RoomId,list\_RDR)

##### □ 操作契约

系统事件	<b>PrintRD(RoomId,list_RDR)</b>
交叉引用	出详单
前置条件	详单对象已经被创建
后置条件	详单文件对象被创建

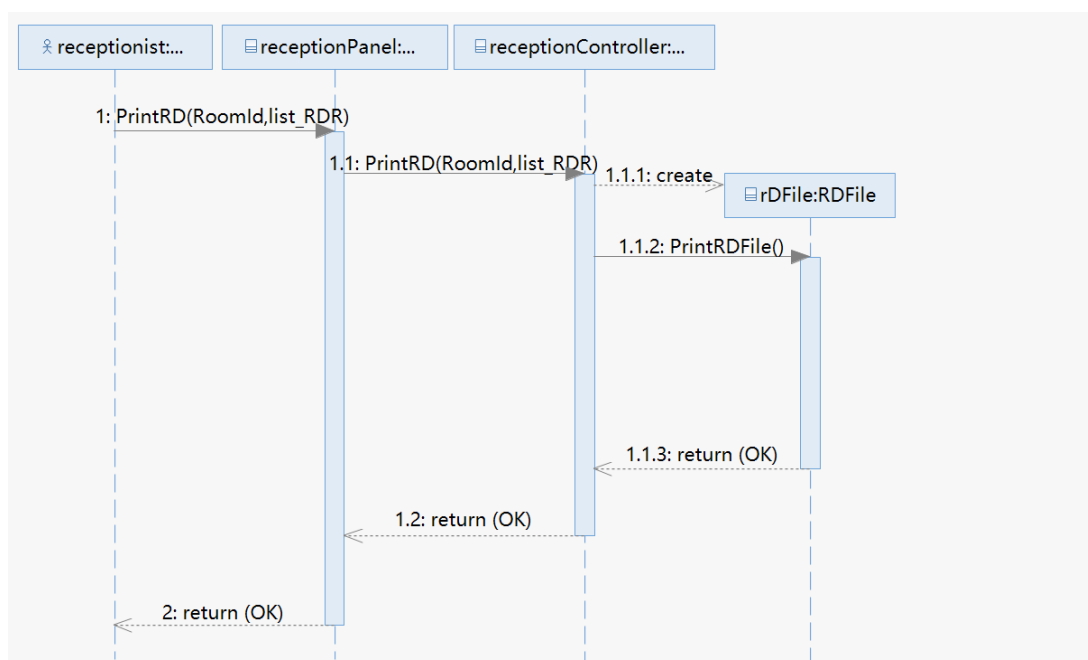
##### □ 设计用例实现过程

1.前台人员点击操作前台面板（ReceptionPanel）

2.前台面板（ReceptionPanel）向前台控制器（ReceptionController）发送  
打印详单请求

### 3.前台控制器（ReceptionController）创建详单文件（RDFile）

#### □ Sequence Diagram



### 3.4.2. 出账单

#### ■ CreateInvoice(RoomId,date\_in,date\_out)

##### 操作契约

系统事件	CreateInvoice(RoomId,date_in,date_out)
交叉引用	出账单
前置条件	该房间对象已经被创建
后置条件	1.账单对象（概念类）被创建 2.账单对象与房间对象建立关联 3.账单对象（概念类）的属性被赋值：房间 i d，入住时间，退房时间，总消费

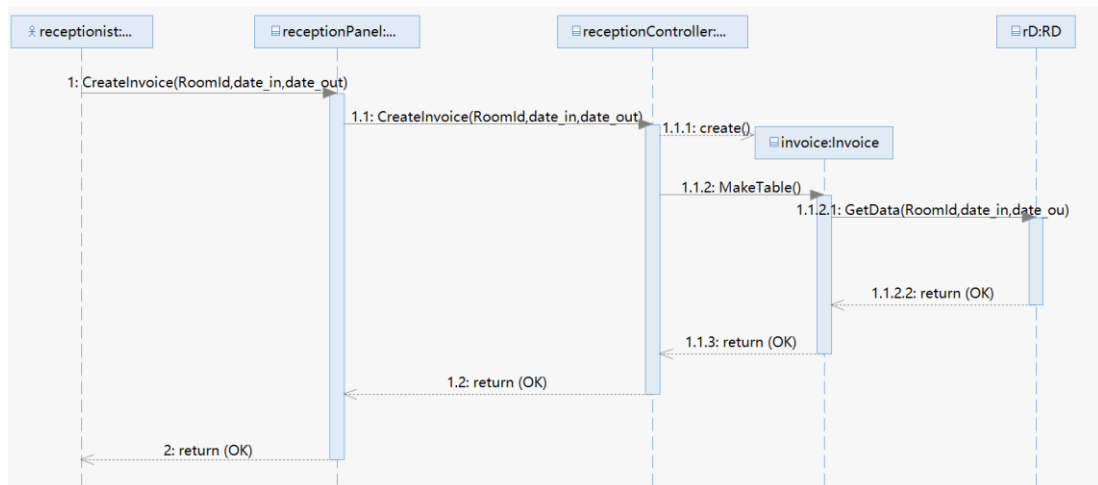
#### □ 设计用例实现过程

##### 1.前台人员点击操作前台面板（ReceptionPanel）

2.前台面板 (ReceptionPanel) 向前台控制器 (ReceptionController) 发送  
创建账单请求

3.前台控制器 (ReceptionController) 创建账单 (Invoice) , 从详单获取数  
据。

#### □ Sequence Diagram



#### ■ PrintInvoice(RoomId,date\_out,Toal\_fee)

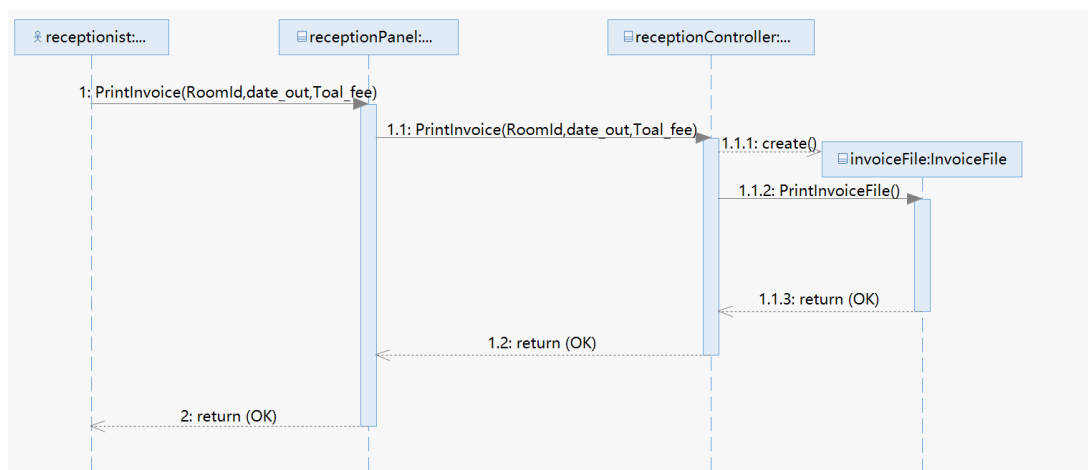
#### □ 操作契约

系统事件	<b>PrintInvoice(RoomId,date_out,Toal_fee)</b>
交叉引用	出账单
前置条件	该账单对象已经被创建
后置条件	账单文件对象被创建

#### □ 设计用例实现过程

- 1.前台人员点击操作前台面板 (ReceptionPanel)
- 2.前台面板 (ReceptionPanel) 向前台控制器 (ReceptionController) 发送  
打印账单请求
- 3.前台控制器 (ReceptionController) 创建账单文件 (InvoiceFile)

#### □ Sequence Diagram



## 3.5. 酒店经理

### 3.5.1. 查看报表

- **QueryReport(list\_RoomId,type\_Report,date)——>Return**  
**(RoomId,Times\_OnOff,Duration,Total\_Fee,Number\_of\_DR,**  
**Times\_ChangeTemp,Times\_changeFanSpeed)**

参数：房间号列表，报表类型，日期。

返回值：开关次数，使用空调的时长，总费用，详单数，调温次数，调风次数，房间号。

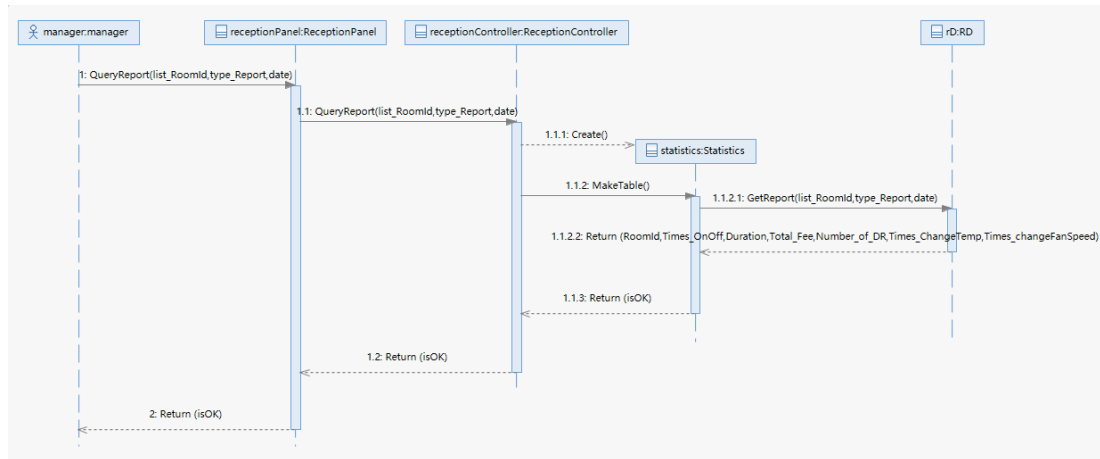
#### □ 操作契约

系统事件	QueryReport(list_RoomId,type_Report,date)
交叉引用	查看报表
前置条件	操作者权限为经理，空调系统正常运行
后置条件	1.统计对象（概念类）被创建 2.统计对象与详单对象建立关联 3.统计对象的各属性被赋值：开关次数，使用空调的时长，总费用，详单数，调温次数，调风次数。

#### □ 设计用例实现过程

- 1.经理选择查看报表。
- 2.前台控制器对象接收信号，创建统计对象，查询详单，接受返回值。
- 3.QueryReport 过程完成。

#### □ Sequence Diagram



#### ■ PrintReport(ReportId,date)-> Return (isOk)

参数：报表号，日期。

返回值：无

#### □ 操作契约

系统事件	PrintReport(ReportId,date)
交叉引用	打印报表（采取输出文件的方式，对应报表实例在 QueryReport（）中已创建并修改）
前置条件	操作者权限为经理，空调系统正常运行
后置条件	1.报表文件对象被创建

#### □ 设计用例实现过程

- 1.经理选择打印报表。
- 2.前台控制器对象接受信号，创建 report.txt，从对应报表中取值填入。
- 3.PrintReport 过程完成。

#### □ Sequence Diagram

