

**分布式温控系统动态结构设计**



**姓名：于越 刘宇航 唐方柳 李文淇 王岚睿**

**班级： 2018211304**

**组号：304d**

**2021年 5 月 20 日**

**文件修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **修订人** | **修订日期** | **修订内容** |
| 1.0 | 于越 刘宇航 王岚睿  李文淇 唐方柳 | 2021-05-20 | 创建、编写文件初稿 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 文档概述 4](#_Toc72774226)

[2 系统及软件架构说明 4](#_Toc72774227)

[2.1 系统说明 4](#_Toc72774228)

[2.2 软件架构 5](#_Toc72774229)

[3 动态结构 6](#_Toc72774230)

[3.1 客户 6](#_Toc72774231)

[3.1.1 PowerOn空调开机 7](#_Toc72774232)

[3.1.2 RequestState查询房间状态 10](#_Toc72774233)

[3.1.3 ChangeTargetTemp更改目标温度 11](#_Toc72774234)

[3.1.4 ChangeFanSpeed更改风速 11](#_Toc72774235)

[3.1.5 RequestRelease(RoomId)释放温控请求 14](#_Toc72774236)

[3.1.6 RequestTempUp()请求升温 15](#_Toc72774237)

[3.1.7 PowerOff()空调关机 15](#_Toc72774238)

[3.2 前台 17](#_Toc72774239)

[3.2.1 CreateInvoice创建账单 18](#_Toc72774240)

[3.2.2 PrintInvoice打印账单 19](#_Toc72774241)

[3.2.3 CreateRd创建详单 19](#_Toc72774242)

[3.2.4 PrintRD打印详单 20](#_Toc72774243)

[3.3 经理 21](#_Toc72774244)

[3.3.1 QueryReport查看报表 21](#_Toc72774245)

[3.3.2 PrintReport 打印报表 22](#_Toc72774246)

[3.4 管理员 23](#_Toc72774247)

[3.4.1 PowerOn打开从控机电源 24](#_Toc72774248)

[3.4.2 setPara设置参数 24](#_Toc72774249)

[3.4.3 StartUp启动空调 25](#_Toc72774250)

[3.4.4 CheckRoomState检查空调状态 26](#_Toc72774251)

[4 关于调度 27](#_Toc72774252)

[5 文档分工 29](#_Toc72774253)

# 1 文档概述

本文档以《BUPT廉价酒店业务场景》为基础，基于《领域模型》和《用例模型》文档给出。本文档描述了BUPT廉价酒店的计费空调系统的动态结构设计。本文档的目的是使甲方尽可能清晰地了解我方设计的软件结构，从而给出建设性建议；使软件的开发人员对软件的动态结构有所了解，统一开发规范，提高开发效率。

本文档的研究对象范围为BUPT廉价酒店的分布式计费空调系统，其中的角色有客户（空调房间分控）、酒店前台、酒店经理、空调管理员，并考虑了调度策略。

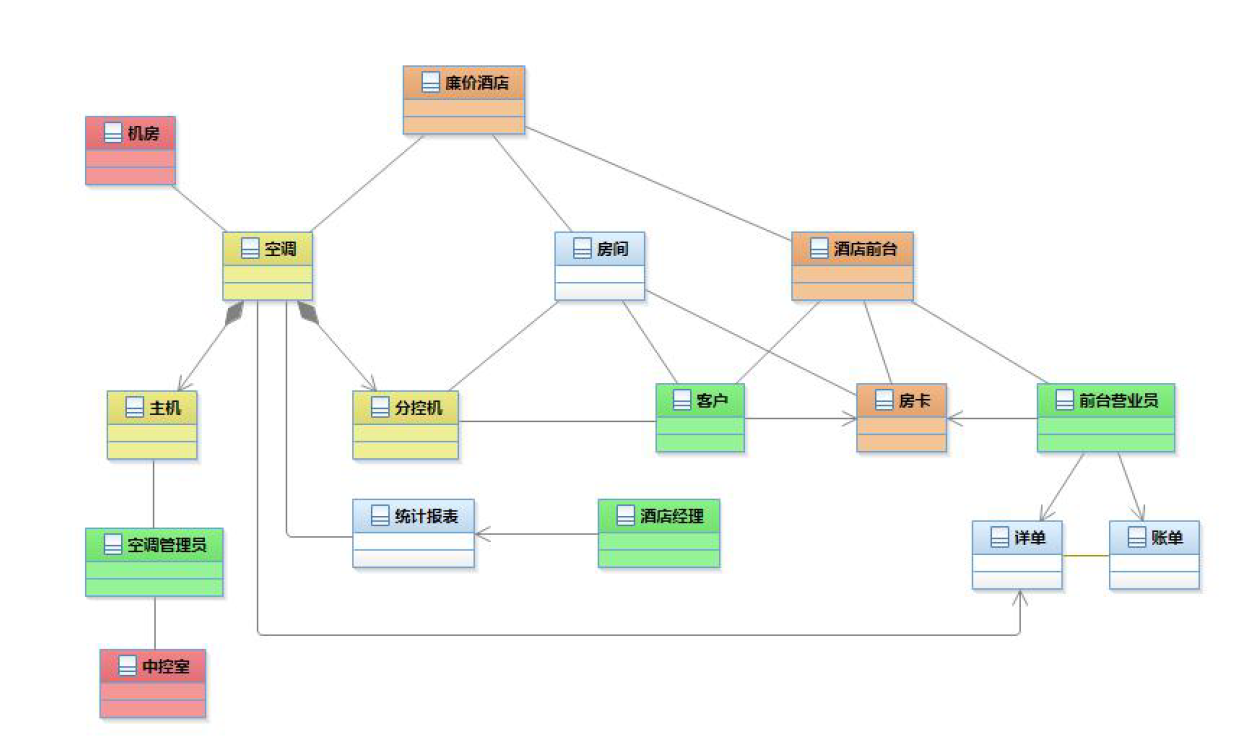
本文档中的动态结构使用UML语言进行描述，使用RSA8.0.3软件进行设计。

# 2 系统及软件架构说明

## 2.1 系统说明

业务场景简介如下：某快捷廉价酒店响应节能绿色环保理念，推行自助计费式中央温控系统，使得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。空调运行期间，酒店的空调管理员能够监控各房间空调的使用状态；需要的情况下可以生成酒店及房间的空调使用统计报表。

领域模型如下，说明了系统中的各个类及其关系：



由此，动态结构中的各个类及其类名称的对应关系如下：

|  |  |
| --- | --- |
| 类 | 动态结构中的类名 |
| 客户 | Customer |
| 前台 | Reception |
| 管理员 | Manager |
| 经理 | Admin |
| 管理员空调控制器 | AcController |
| 监控器 | Monitor |
| 统计数据 | Statistic |
| 报表 | Report |
| 顾客控制空调的控制器 | CustomerController |
| 详单 | RDFile |
| 计时器 | Timer |
| 调度器 | Scheduler |
| 服务队列 | ServiceQueue |
| 等待队列 | WaitQueue |

## 2.2 软件架构

本软件系统采用分层结构，分为用户界面层（前端UI）、控制层、应用逻辑层、持久化层、数据库。由于持久化层的操作比较基础和相似，因此文档中并没有在动态结构的图中出现。

用户界面层：负责与用户的交互，具体体现为客户端的GUI界面。GUI中，每个功能需求都有其对应的交互组件供用户使用。用户界面层将用户的命令发送给控制层对应的Controller进行解析。

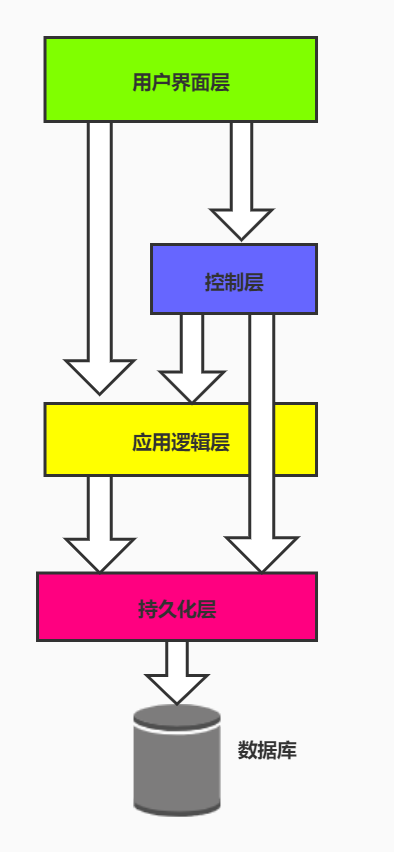
控制层：负责用户界面层发来的命令的处理，具体体现为后端的Controller部分。控制层接收到UI层的命令后进行解析（具体实现为解析json字符串），经过一定的逻辑判断处理，将控制信息传给应用层。

应用逻辑层：具体业务逻辑的实现，体现为后端的大部分处理。接收到控制层具体的控制信息后，应用逻辑层进行对应功能的实现，并将结果传回。有时还需要将结果传至持久化层进行持久化。

持久化层：负责进行数据的存储。应用逻辑层传来要存储的数据后，进行数据的结构化，之后连接数据库，将数据存入数据库。

数据库：存储数据的地方。采用结构化数据库。

示意图如下：

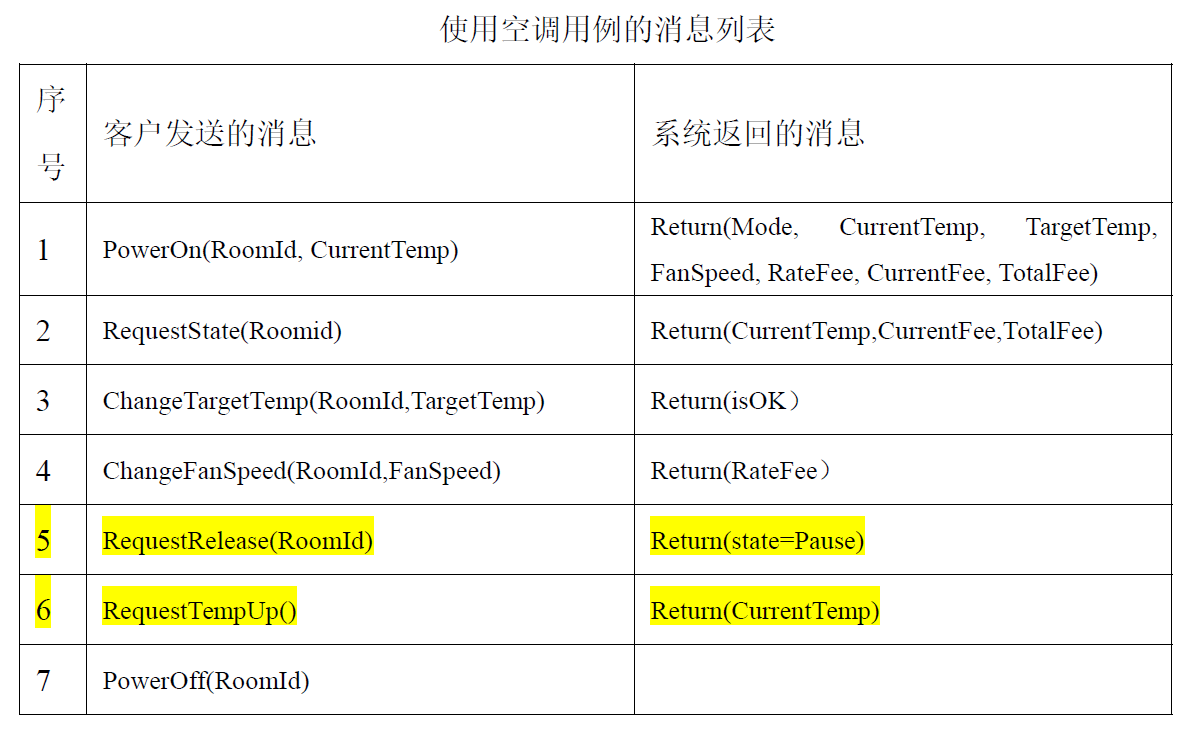


# 3 动态结构

## 3.1 客户

**(注：调度中与无调度中的系统消息相同时，不再标注相同部分，详细的在无调度图中)**

消息列表：

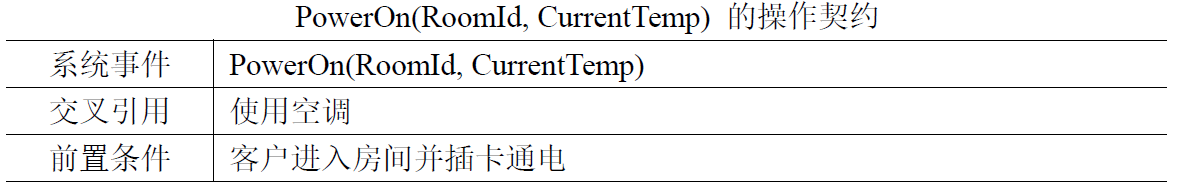


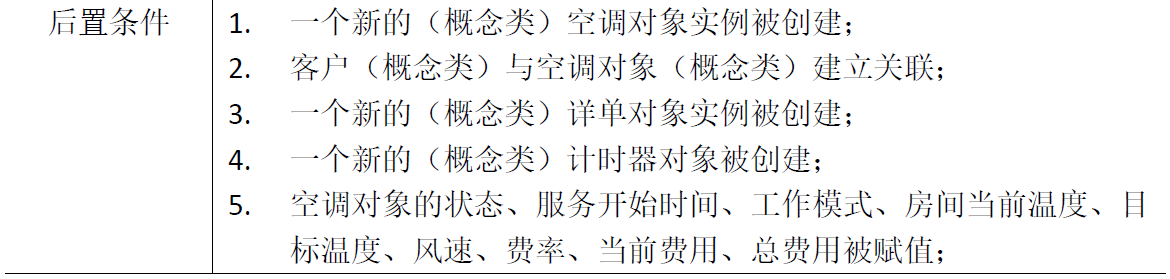
### 3.1.1 PowerOn空调开机

角色：用户

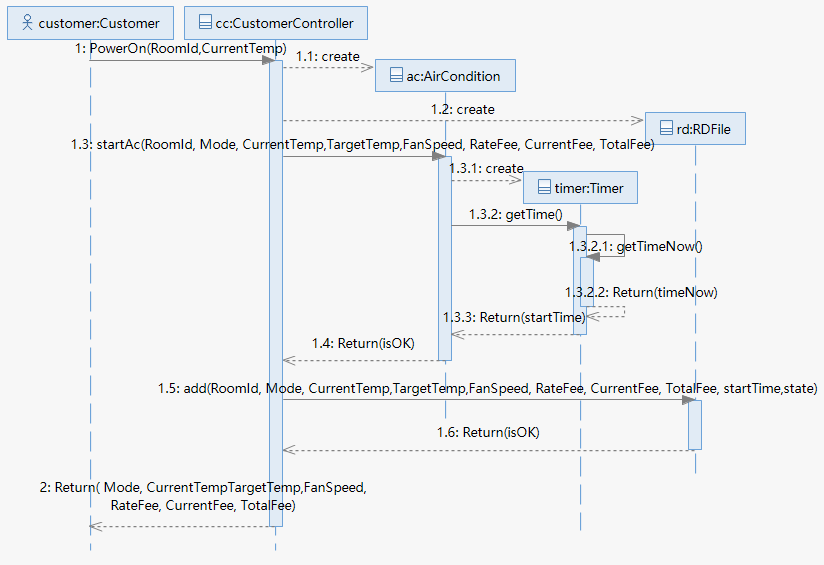
用例：使用空调

功能：客户将空调从控机





无调度：



1.空调开机，设定房间当前温度CurrentTemp

1.1 创建空调对象

1.2 创建详单对象

1.3 启动空调，将参数房间号RoomId，工作模式Mode，房间当前温度CurrentTemp，设定温度TargetTemp，风速FanSpeed，费率FeeRate，当前费用CurrentFee，总费用TotalFee

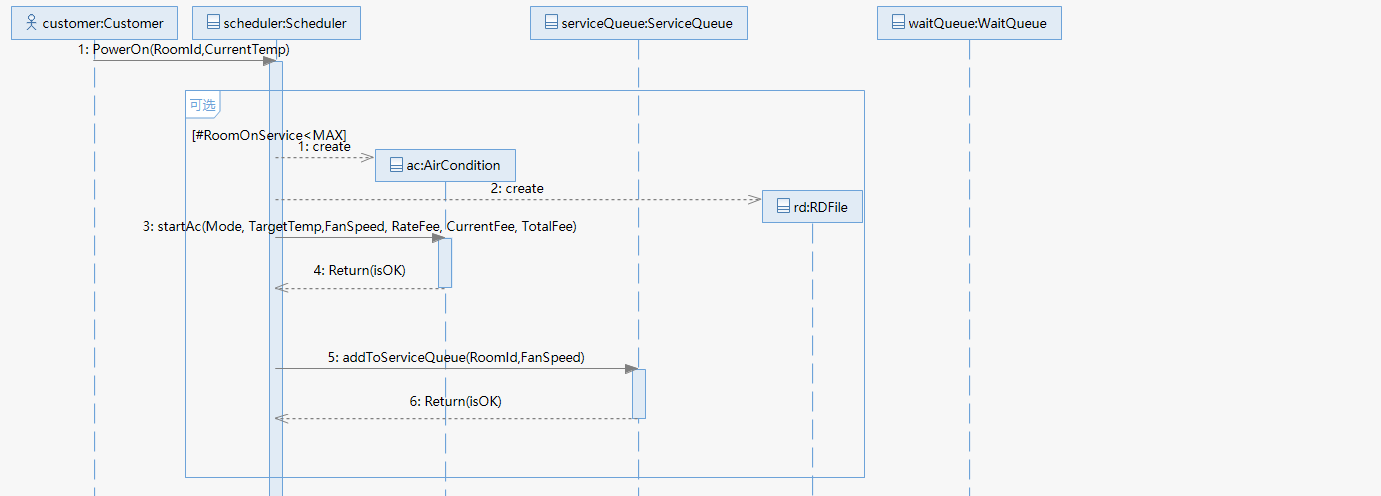
1.3.1 创建房间的计时器对象Timer

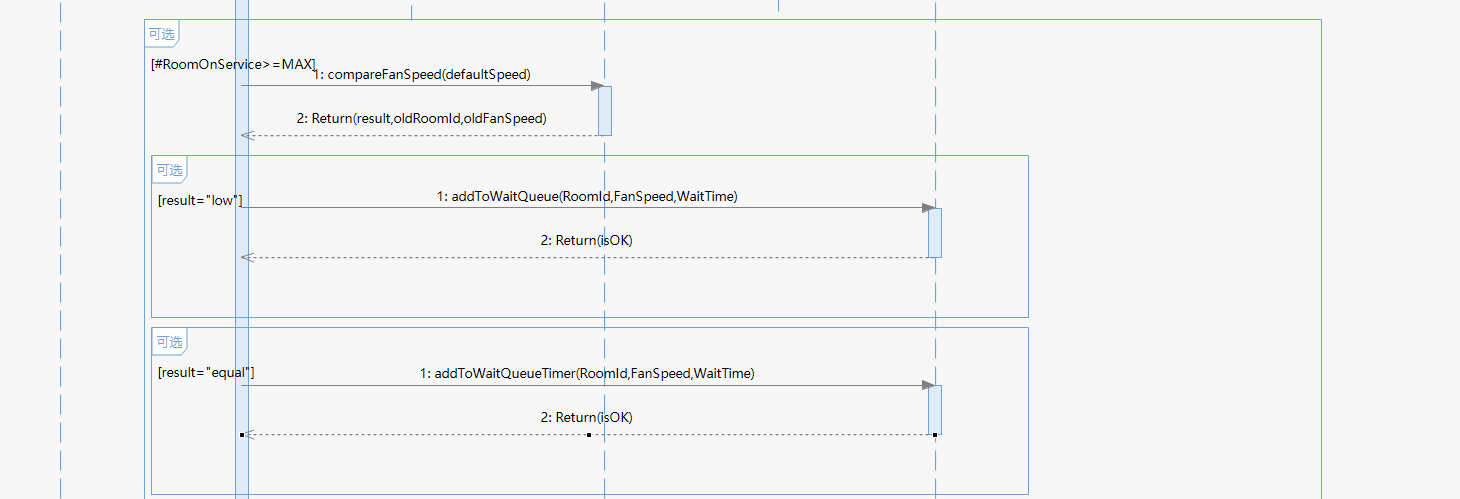
1.3.2 从Timer获取房间启动的时间

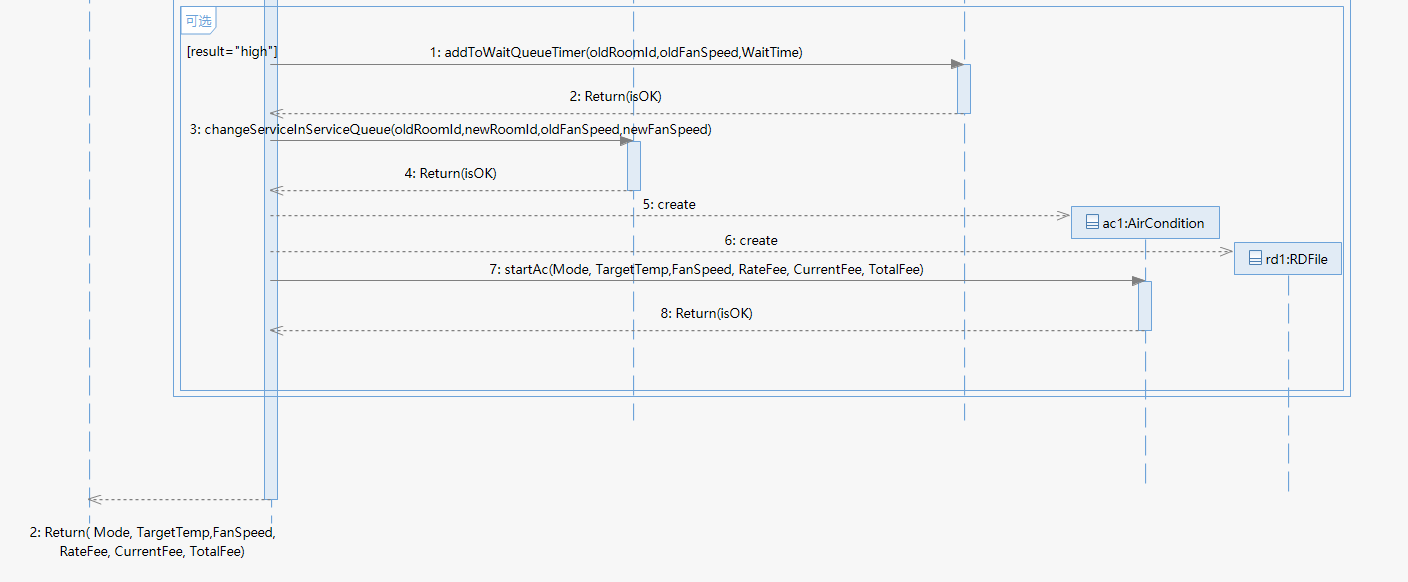
1.5 将上述信息加入详单的一条记录

2 将上述信息返回前端显示

**调度：**







1 向调度器发送启动空调请求

[正在接受服务的空调数量小于设定的最大值]

1 调度器创建空调对象

2 调度器创建详单对象

3 调度器发送启动空调命令

5 调度器将空调加入服务队列，传送的参数有RoomId房间号、FanSpeed风速

[正在接受服务的空调数量大于或等于设定的最大值]

1 比较当前请求的风速（defaultSpeed）与服务队列中各个风速的情况

2 返回比较结果，result为结果，oldRoomId和oldFanSpeed是会被替换的房间的房间号和风速（result不是low的情况）

[当前请求的风速比服务队列中的所有风速都低]

1 调度器设定一个该请求的等待时间，将请求的房间号、风速、等待时间加入等待队列等待定时触发**（以下简称“加入等待队列”，只要加入等待队列都会分配一个等待时长，等待时长到达后会与服务对象进行比较，决定继续等待还是加入服务队列）**

[当前请求的风速与服务队列中的所有风速都相等]

1 将请求加入等待队列

[存在一个服务队列中的风速比请求风速低]

1 调度器将需要被替换的加入等待队列

3 调度器将请求加入服务队列

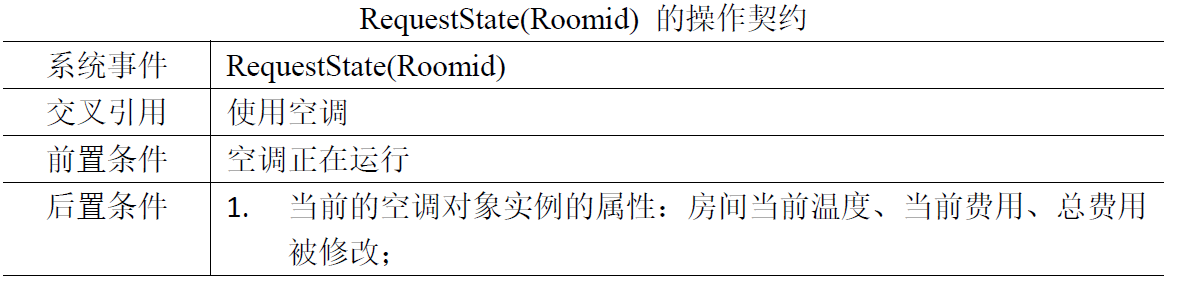
后面的与无调度时一样

### 3.1.2 RequestState查询房间状态

角色：用户

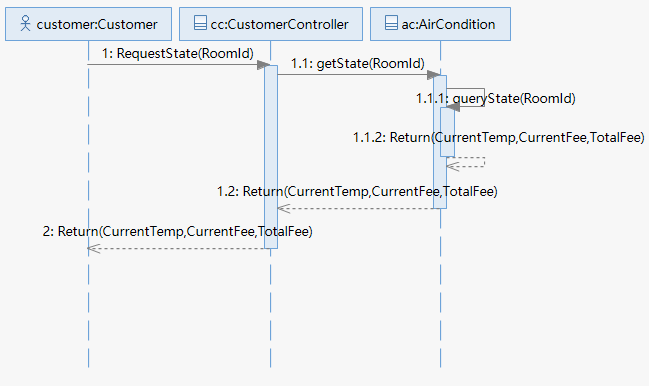
用例：使用空调

功能：查询空调状态，用于显示在面板上



->: RequestState(RoomId)

<-: Return(CurrentTemp,CurrentFee,TotalFee)



1 根据房间号向控制器查询房间号为RoomId的房间的状态

1.1 向空调对象查询状态

1.1.1 查询空调状态

1.1.2 返回当前温度CurrentTemp，当前费用（本次开机以来）CurrentFee以及总费用TotalFee

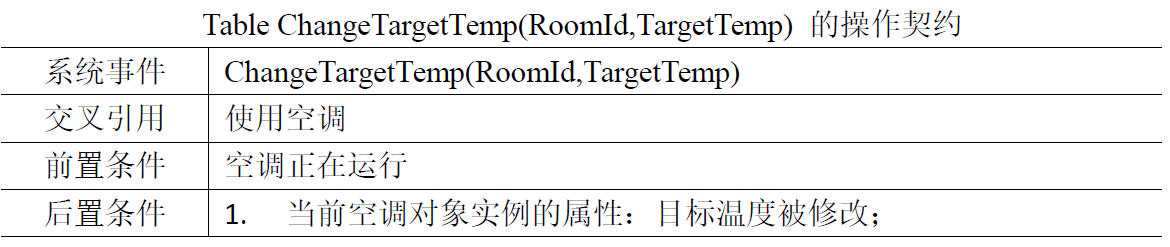
1.2 2 同上

### 3.1.3 ChangeTargetTemp更改目标温度

角色：客户

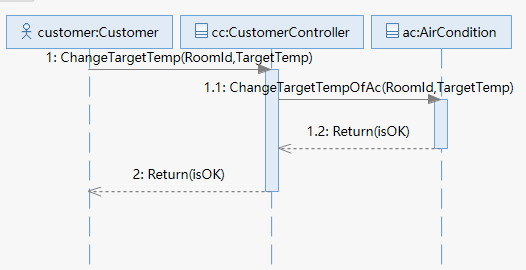
用例：使用空调

功能：更改空调的目标温度



->: ChangeTargetTemp(RoomId,TargetTemp)

<-: Return(isOK)



1 向控制器发送更改房间目标温度的请求，参数为房间号RoomId和设定的目标温度TargetTemp

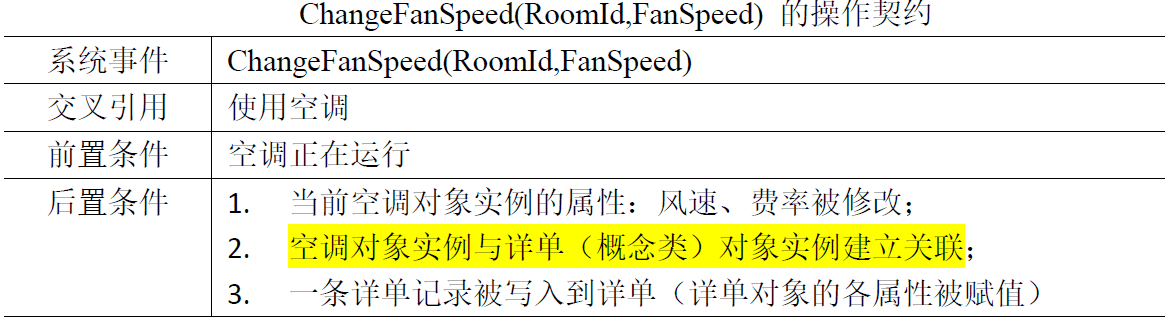
1.1 向空调对象发送更改房间目标温度的请求

### 3.1.4 ChangeFanSpeed更改风速

角色：用户

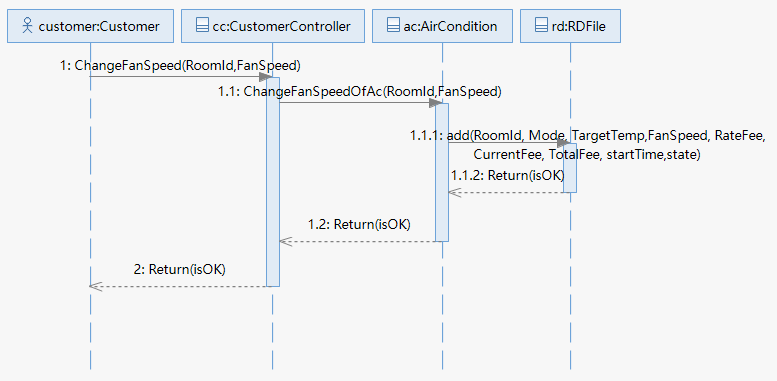
用例：使用空调

功能：更改空调风速



->: ChangeFanSpeed(RoomId,FanSpeed)

<-: Return(isOK)

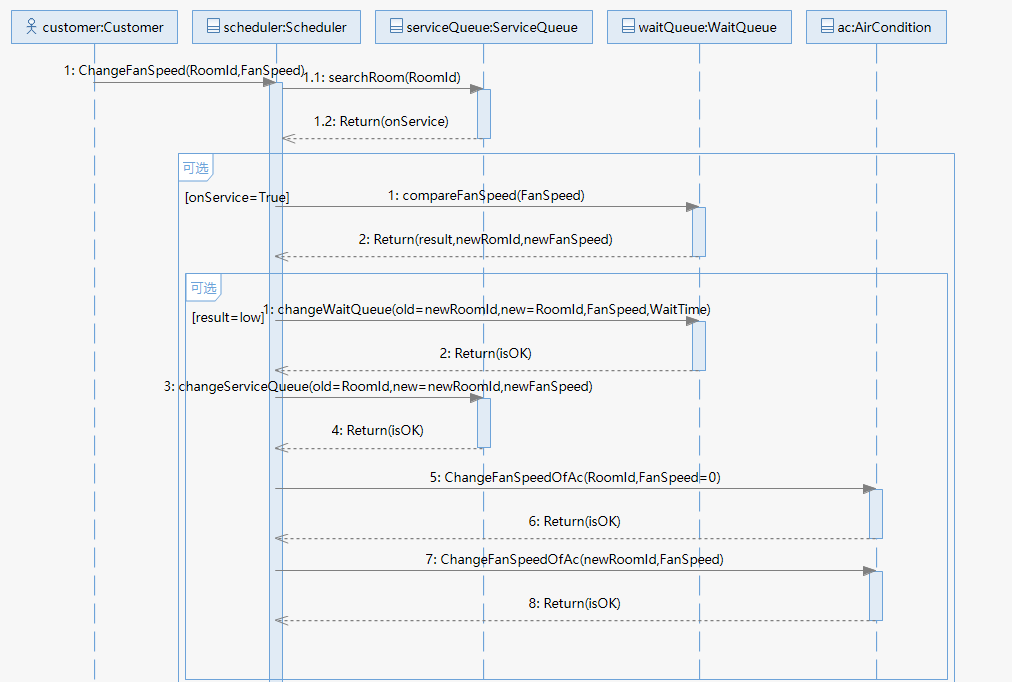


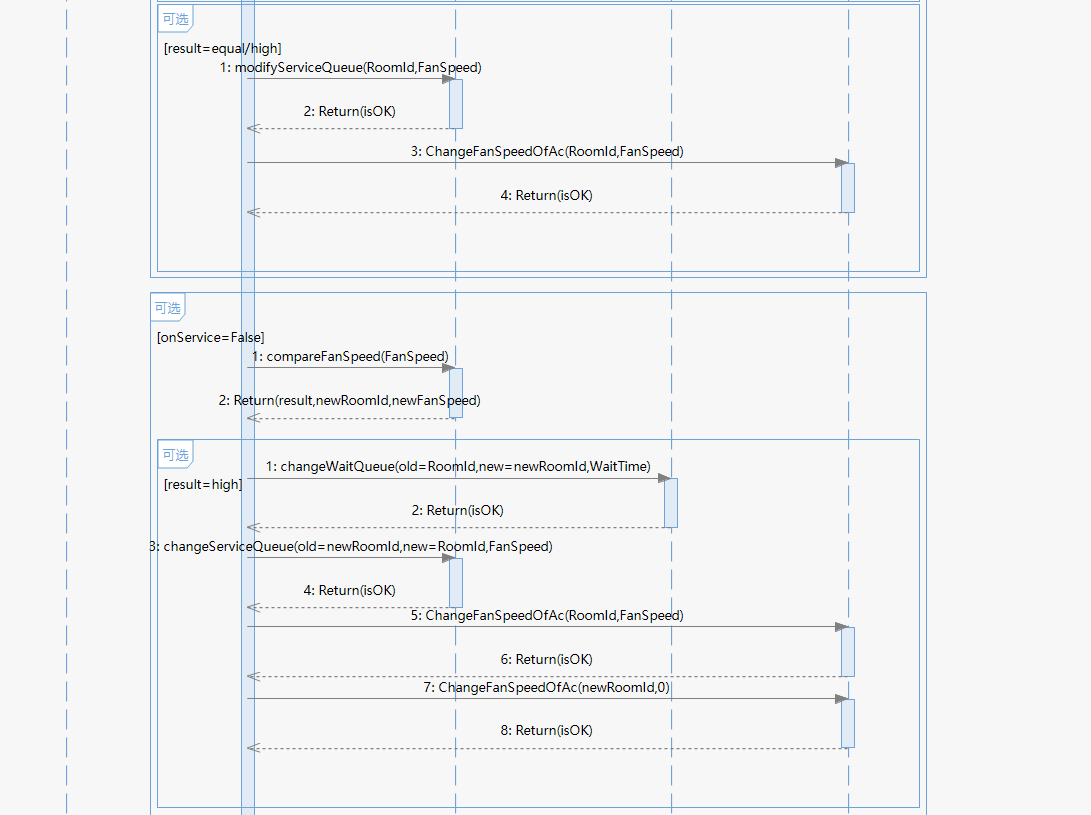
1 向控制器发送改变房间风速请求，参数为房间号RoomId和风速FanSpeed

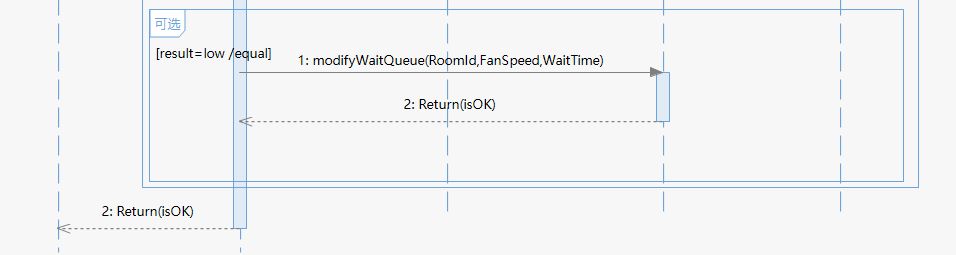
1.1 向空调对象发送改变风速请求

1.1.1 将更改后的信息发送给详单对象存储

**调度：**







1 向调度器发送改变风速命令

1.1 在服务队列中查找RoomId

1.2 返回房间是否在服务队列onService

[如果房间在服务队列中]

1 比较请求的FanSpeed与等待队列中各个风速的大小

2 返回结果

[如果比等待队列之中的至少一个小，选出最适合替换的，下称“新房间”]

1 将等待队列中的新房间及风速替换为请求房间及风速

3 将服务队列中的请求房间及风速替换为新房间及风速

5 将请求房间的风速设置为0

7 将新房间的风速设置为新风速

[如果大于或等于等待队列中的风速]

1 更改服务队列中的请求房间的风速为FanSpeed

3 将请求房间的风速改为FanSpeed

[如果房间不在服务队列中]

1 比较请求风速与服务队列中各风速的大小

[如果存在至少一个比请求风速小的，选出最适合被替换的，称为“新房间”]

1 将等待队列中的请求房间替换为新房间及风速

3 将服务队列中的新房间替换为请求房间及风速

5 将请求房间的风速设为FanSpeed

7 将新房间的风速设为0

[如果请求风俗小于或等于服务队列中的风速]

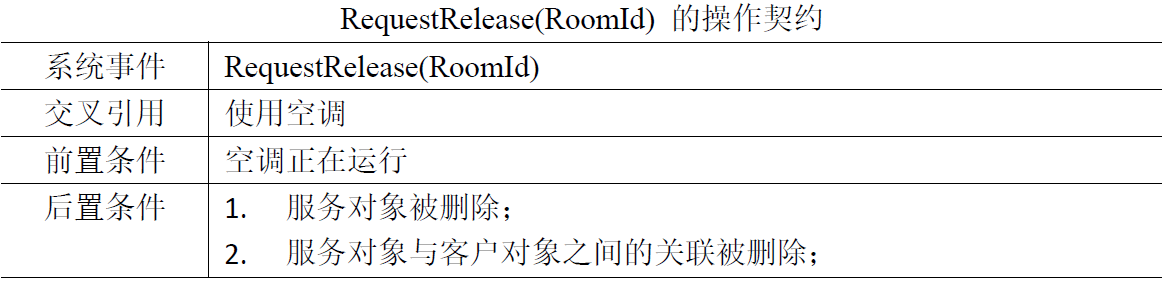
1 更改等待队列中的请求房间的风速及时间片

### 3.1.5 RequestRelease(RoomId)释放温控请求

角色：客户

用例：使用空调

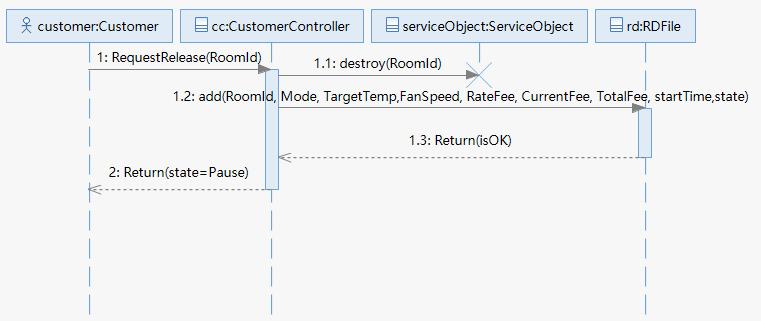
功能：到达室温之后停止升温





->: RequestRelease(RoomId)

<-: Return(state=Pause)



1 向控制器发送释放服务对象的请求

1.1 释放服务对象

1.2 将记录加入详单

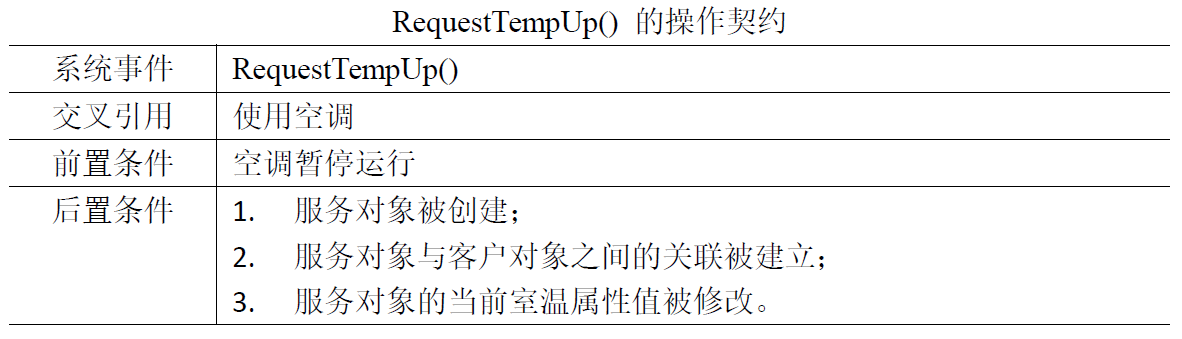
2 返回房间状态为暂停

### 3.1.6 RequestTempUp()请求升温

角色：客户

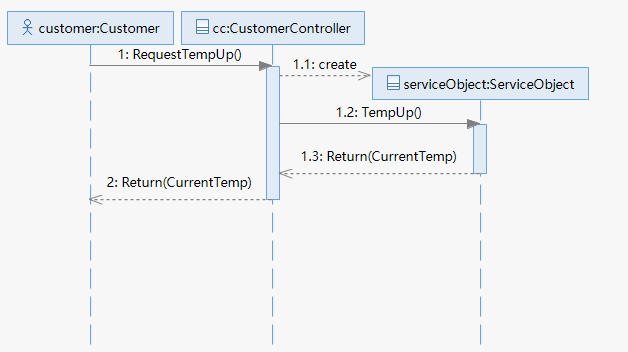
用例：使用空调

功能：使当前房间的温度+1



->: RequestTempUp()

<-: Return(CurrentTemp)



1 向控制器发送升温的请求

1.1 创建负责调整温度的服务对象

1.2 控制器向服务对象发送升温请求

1.3 服务对象调整温度之后将现在的温度CurrentTemp返回

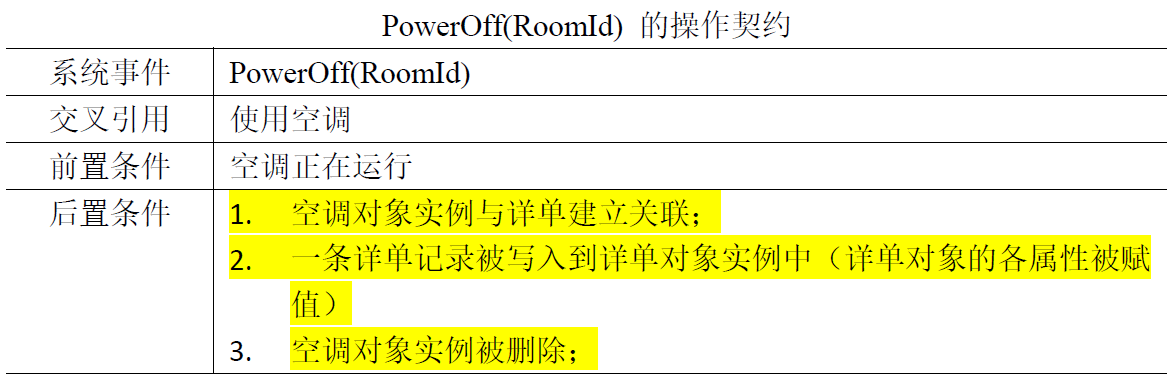
2 返回当前温度

### 3.1.7 PowerOff()空调关机

角色：客户

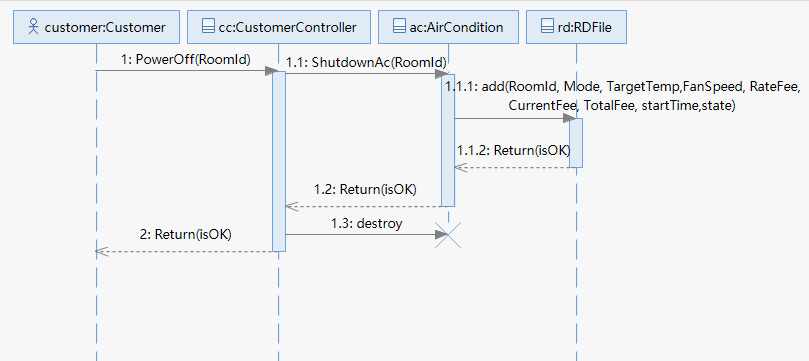
用例：使用空调

功能：关闭空调从控机



->: PowerOff(RoomId)

<-: Return(isOK)



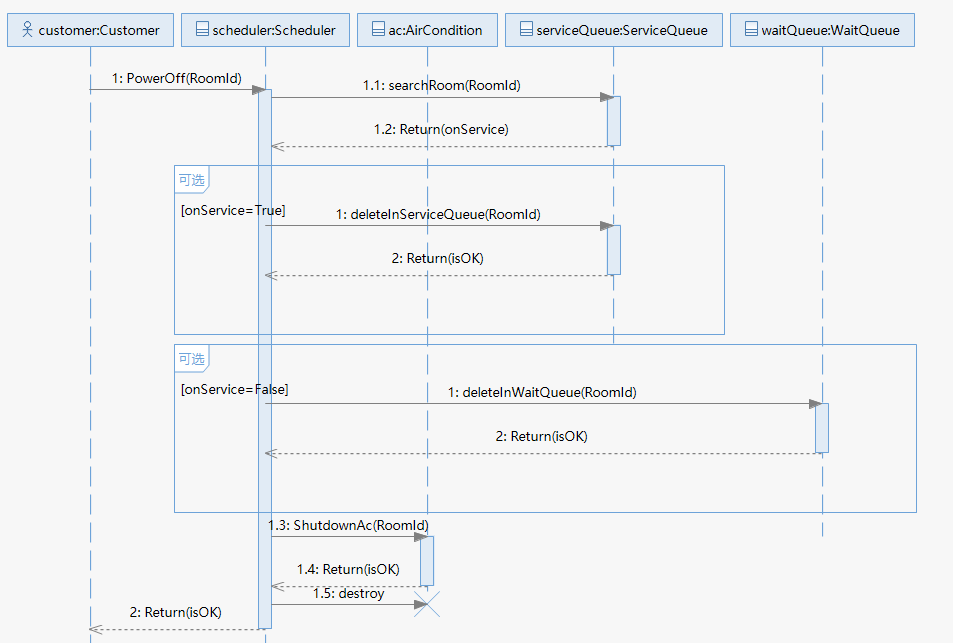
1 向控制器发送关机请求，指定RoomId

1.1 向空调对象发送关机请求

1.1.1 向详单添加一条记录

2 关机

**调度：**



1 向调度器发送关机指令，参数为RoomId

1.1 在服务队列中查找房间

1.2 返回是否在服务队列中onService

[如果在服务队列中]

1 将服务队列中的房间删掉

[如果在等待队列中]

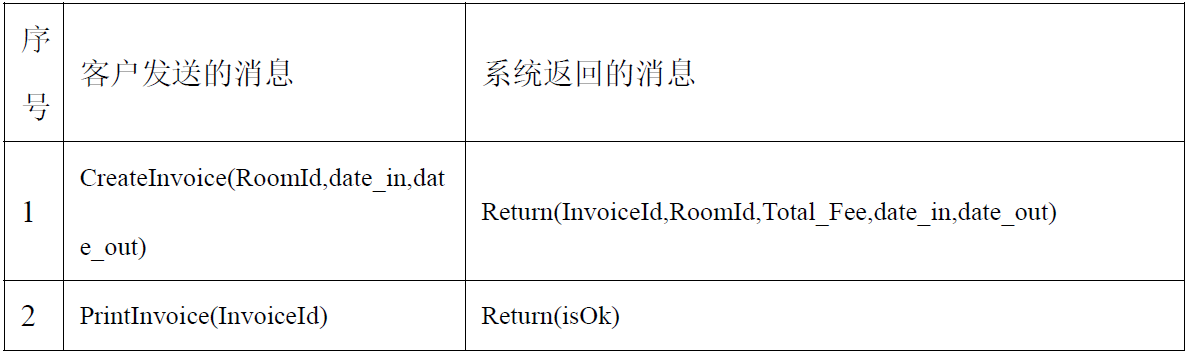
1 将等待队列中的房间删掉

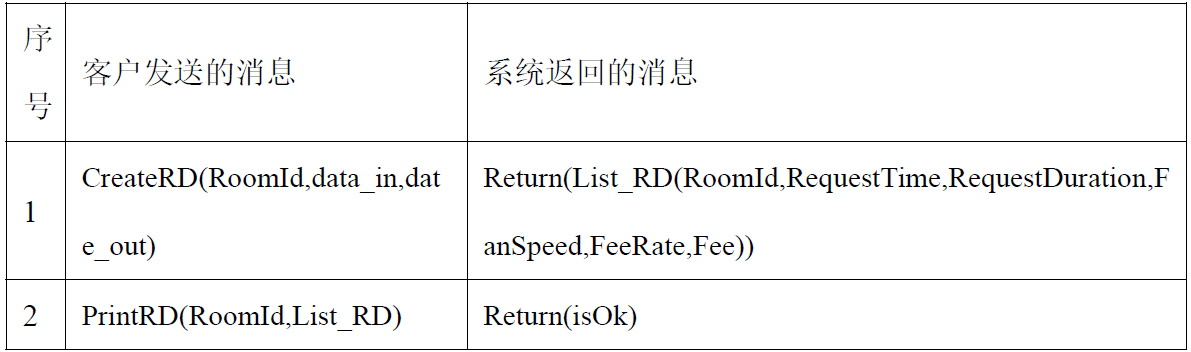
1.3 向空调对象发送关闭空调消息

1.5 释放空调对象

## 3.2 前台

消息列表：



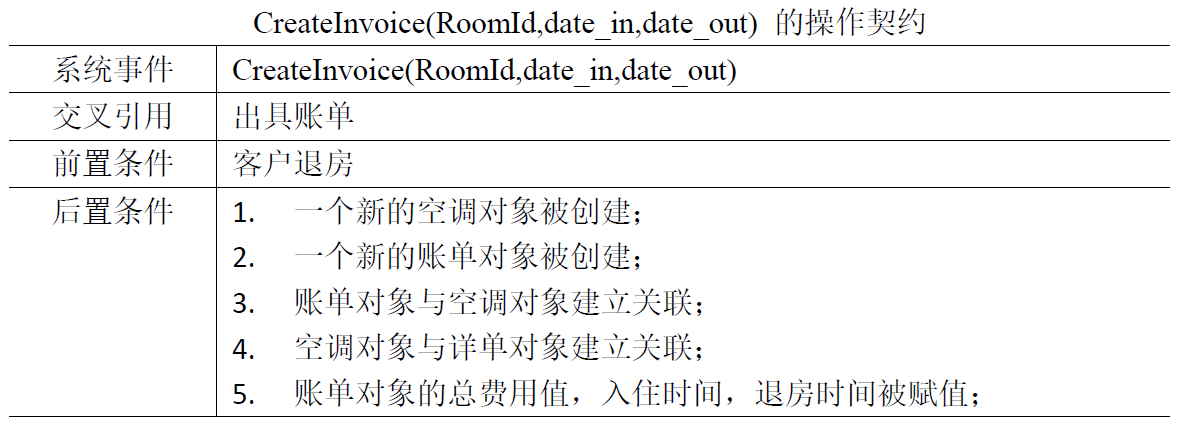


### 3.2.1 CreateInvoice创建账单

角色：前台

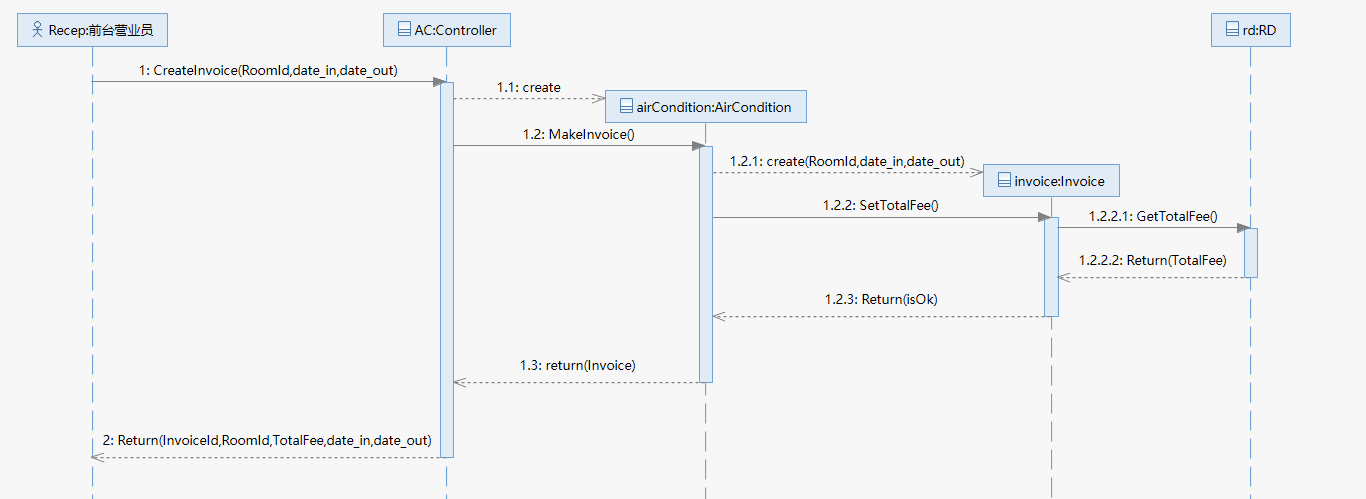
用例：打印账单

功能：结账时前台接待员创建账单



->: CreateInvoice(RoomId, date\_in, date\_out)

<-: Return(InvoiceId,RoodId,TotalFee,date\_in,date\_out)



1: 前台营业员将出具账单的请求发送给控制器层

1.1: 控制器创建一个空调对象

1.2: 控制器将创建账单对象的命令发送给空调对象

1.2.1: 创建一个账单对象并赋值

1.2.2: 通过详单获得总费用并给总费用赋值

1.2.2.1: 与详单建立关联并获得账单总费用

1.2.2.2: 信息获取成功的返回总费用

1.2.3: 赋值成功的返回

1.3: 返回赋值成功的账单对象

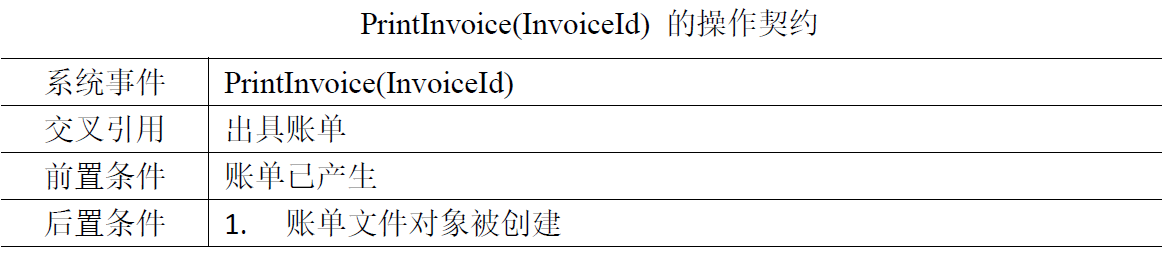
2: 返回账单的ID，房间ID，费用，入住时间和退房时间

### 3.2.2 PrintInvoice打印账单

角色：前台

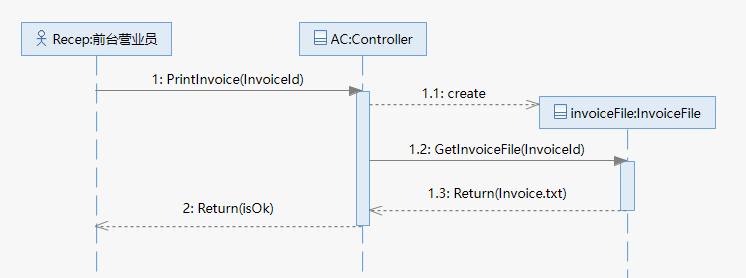
用例：出账单

功能：结账时前台接待员打印账单



->: PrintInvoice(InvoiceId)

<-:Return(isOK)



1: 前台营业员将打印账单的请求发送给控制器层

1.1: 创建一个账单文件对象

1.2: 打印实际账单文件的命令

1.3: 返回实际账单TXT

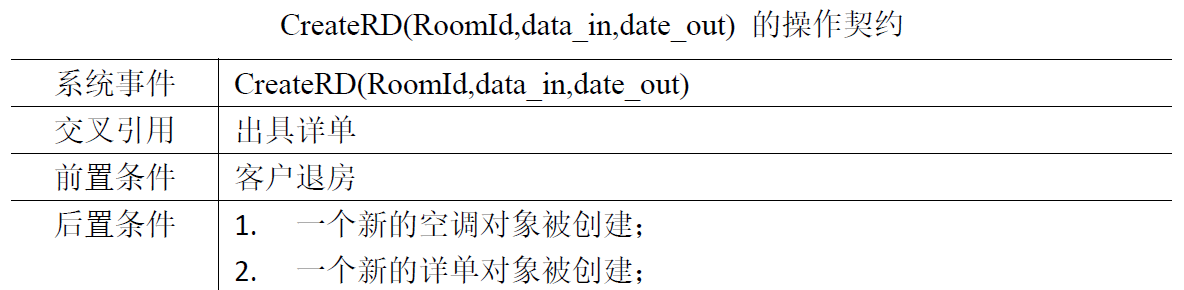
2: 返回打印账单成功的信息

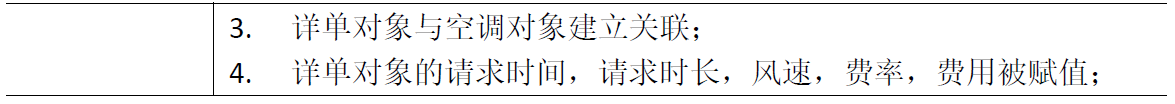
### 3.2.3 CreateRd创建详单

角色：前台

用例：出详单

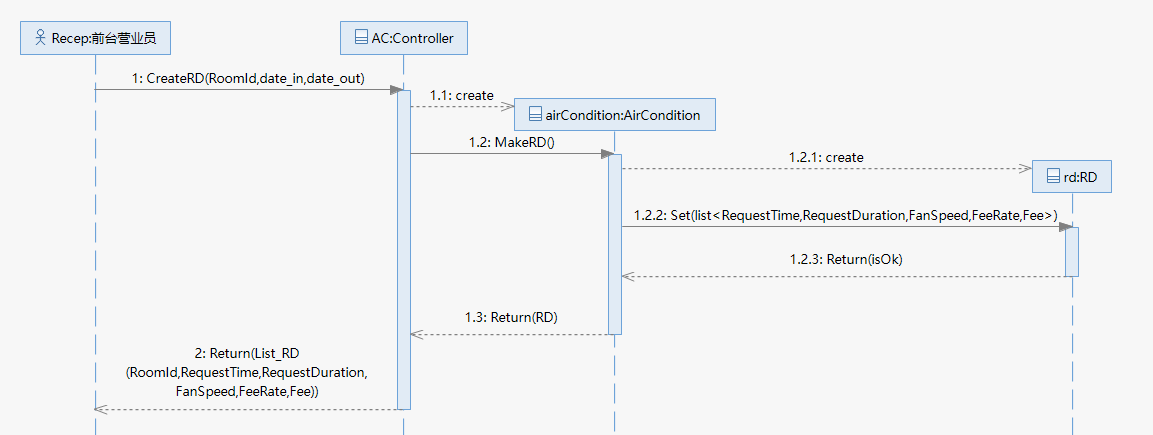
功能：前台接待员在结账时创建详单





->: CreateRD(RoomId,date\_in,date\_out)

<-: Return(List\_RD(RoomId,RequestTime,RequestDuration,FanSpeed,FeeRate,Fee))



1: 前台营业员将出具详单的请求发送给控制器层

1.1: 创建一个空调对象

1.2: 创建详单对象的命令

1.2.1: 创建一个账单对象

1.2.2: 对详单对象进行赋值（参数是一个消费列表）

1.2.3: 赋值成功的返回

1.3: 返回赋值成功的账单对象

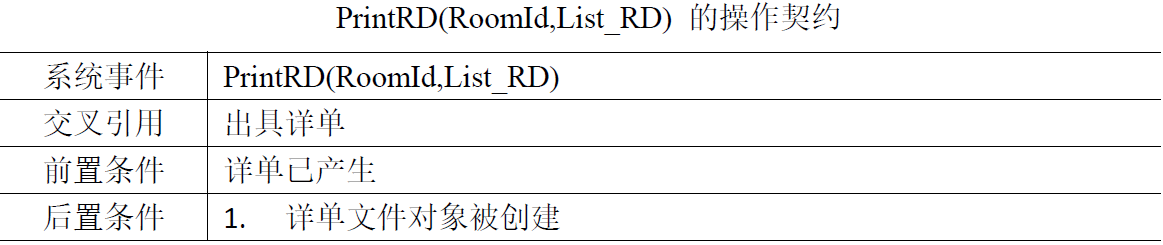
2: 返回房间ID，请求时间，请求时长，风速，费率，费用组成的列表

### 3.2.4 PrintRD打印详单

角色：前台

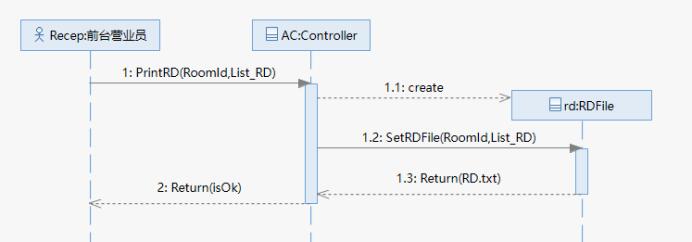
用例：出详单

功能：前台接待员在结账时打印详单



->: PrintRD(RoomId,List\_RD)

<-: Return(isOK)



1: 前台营业员将打印详单的请求发送给控制器层

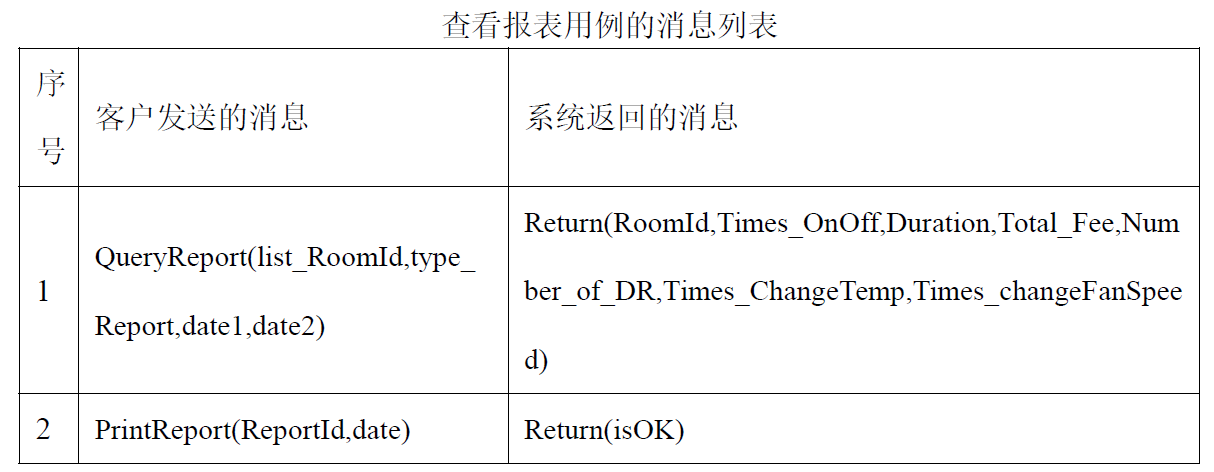
1.1: 创建一个详单文件对象

1.2: 打印实际详单文件的命令

1.3: 返回实际详单TXT文件

2: 返回打印详单成功的信息

## 3.3 经理

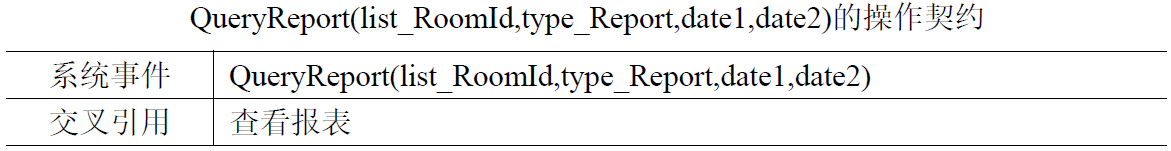


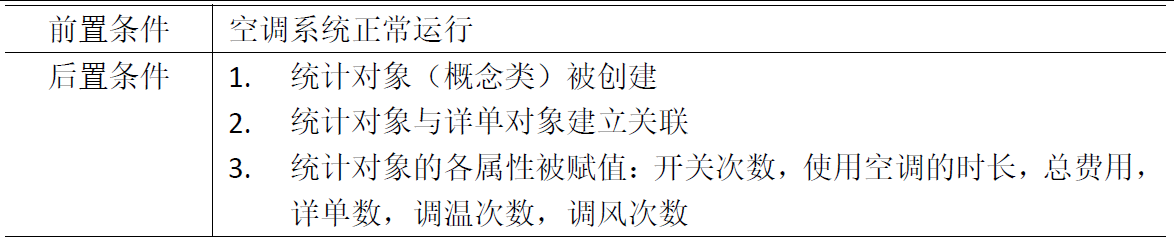
### 3.3.1 QueryReport查看报表

角色：经理

用例：查看报表

功能：查看房间的id，对应的空调开关次数，开启时间，总费用，详单数，调温次数，调风次数

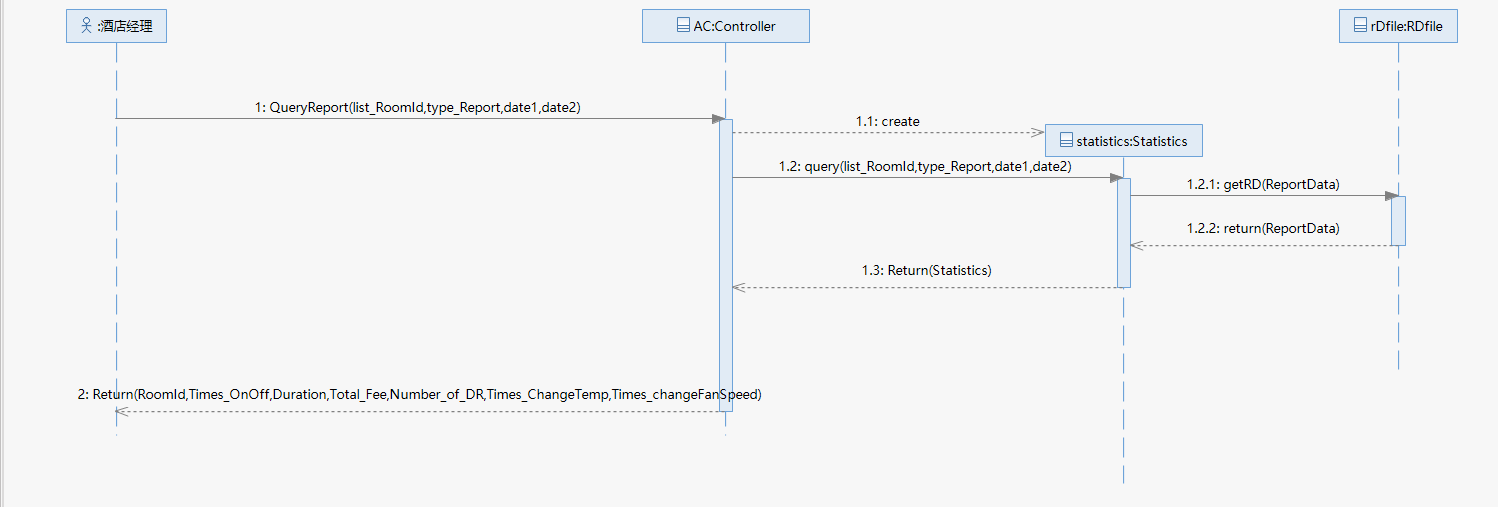




->: QueryReport(list\_RoomId,type\_Report,date1,date2)

<-: Return(RoomId,Times\_OnOff,Duration,Total\_Fee,Number\_of\_DR,

Times\_ChangeTemp,Times\_changeFanSpeed)



1. 经理发送 QueryReport来请求获得报表

1.1 统计对象类被创建

1.2 获得统计信息

1.2.1 获取报表数据ReportData

1.2.2 报表数据被返回

1.3 返回统计数据

2 经理获得详单信息RoomId,Times\_OnOff,Duration,Total\_Fee,Number\_of\_DR,

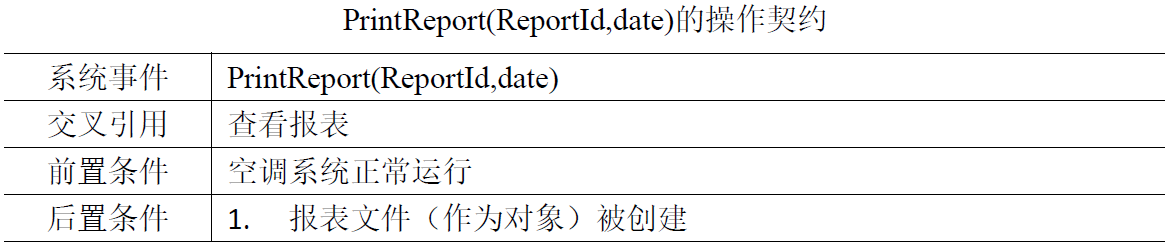
Times\_ChangeTemp,Times\_changeFanSpeed

### 3.3.2 PrintReport 打印报表

角色：经理

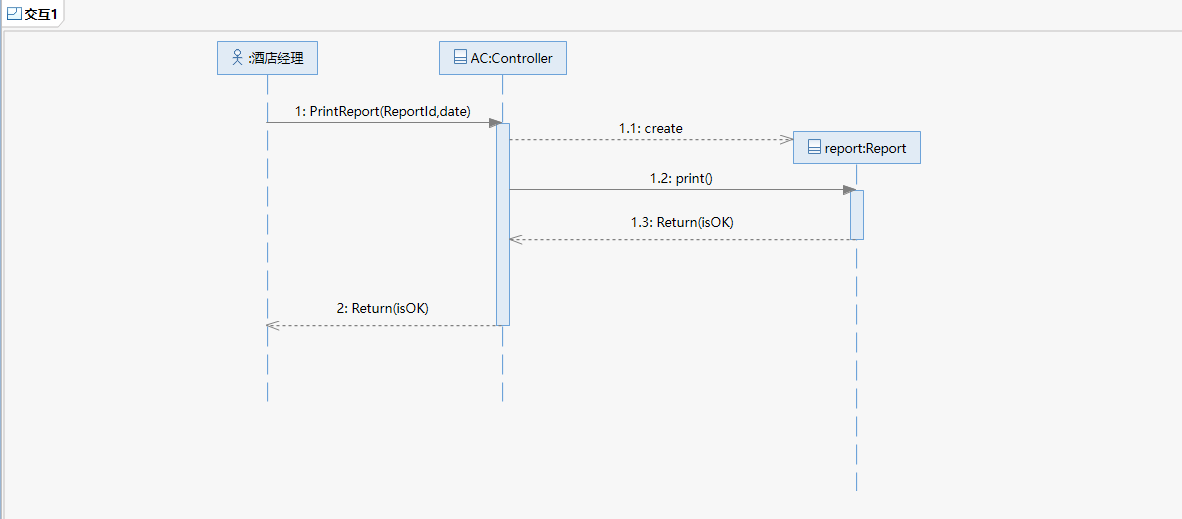
用例：打印报表

功能：打印报表



->: PrintReport(ReportId,date)

<-: Return(isOK)



1.发送打印请求

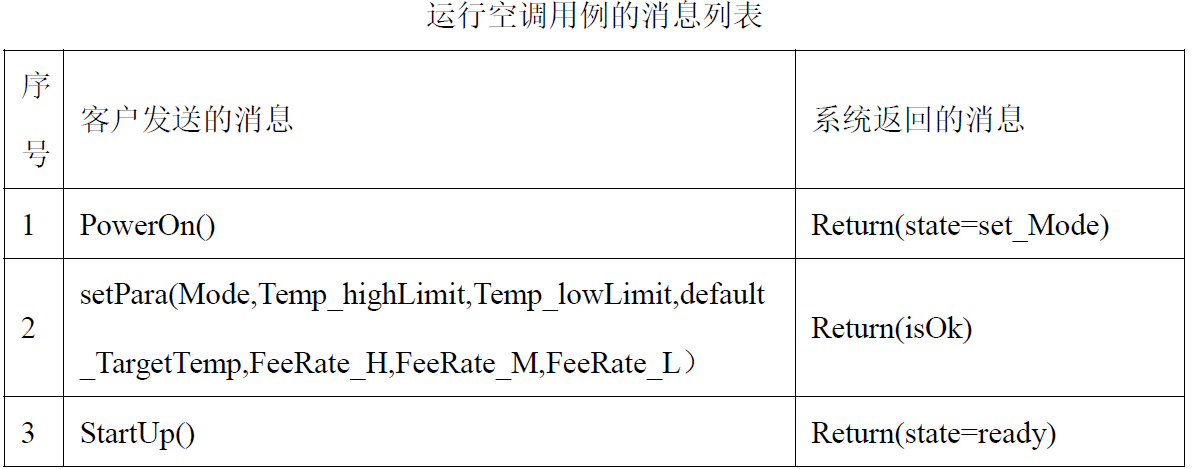
1.1 报表文件被创建

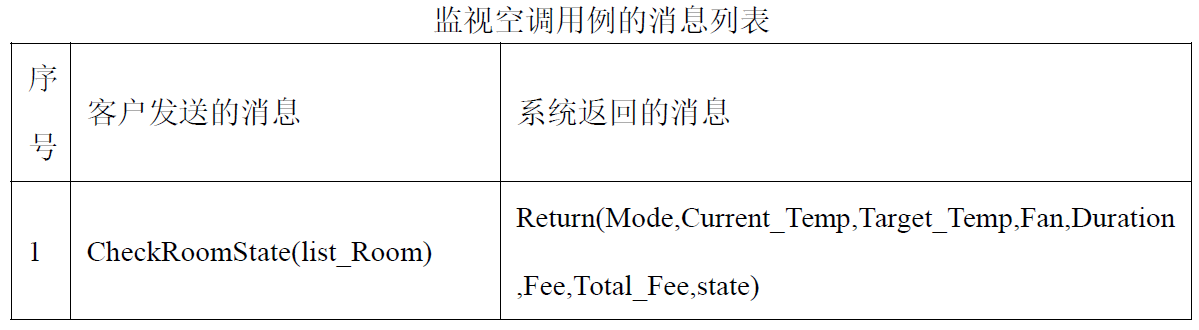
1.2 打印报表请求发送

1.3 成功打印

2 返回成功标志

## 3.4 管理员



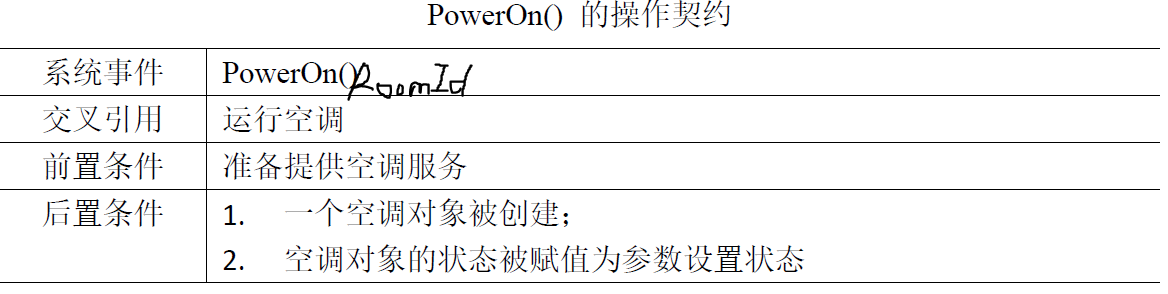


### 3.4.1 PowerOn打开从控机电源

角色：管理员

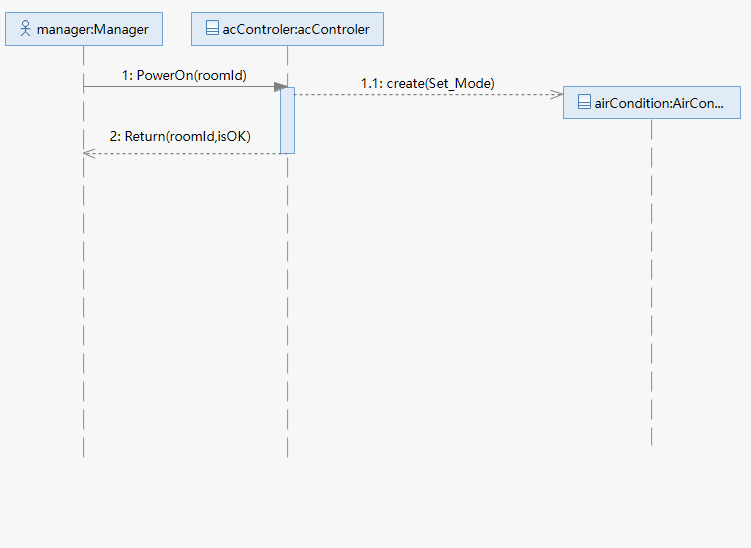
用例：运行空调

功能：管理员打开从控机空调电源，准备进行参数设置。



->: PowerOn(RoomId)

<-: Return(RoomId,isOK)



1. 管理员将打开空调的指令发给空调控制器（打开电源）

1.1. 空调控制器创建空调对象，完成空调对象初始化，状态设置为Set\_Mode

2. PowerOn()完成

### 3.4.2 setPara设置参数

角色：管理员

用例：运行空调

功能：设置空调运行参数

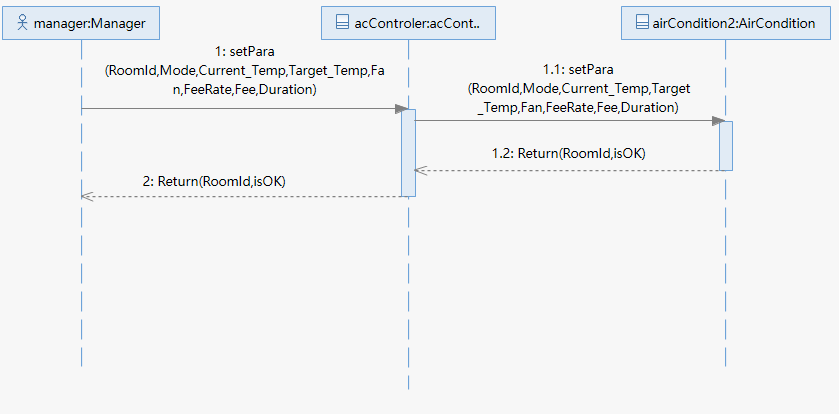
setPara的操作契约

|  |  |
| --- | --- |
| 系统事件 | setPara(Mode,Temp\_highLimit,Temp\_lowLimit,default\_TargetTemp,  FeeRate\_H,FeeRate\_M,FeeRate\_L) |
| 交叉引用 | 运行空调 |
| 前置条件 | 空调对象被创建且工作状态为set\_Mode |
| 后置条件 | 1. 空调对象的工作模式、温控范围、缺省的目标温度、缺省的风速、费率被赋值； |

->: setPara(Mode,Temp\_highLimit,Temp\_lowLimit,default\_TargetTemp,

FeeRate\_H,FeeRate\_M,FeeRate\_L)

<-: Return(RoomId,isOK)



1. 管理员通过空调控制器启动空调服务

1.1 空调控制器设置对应的空调对象的状态属性为：就绪

1.2 空调对象状态设为就绪

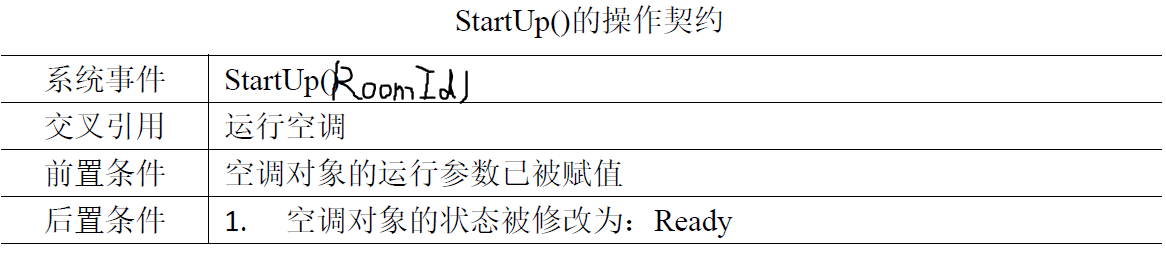
2. 空调启动成功

### 3.4.3 StartUp启动空调

角色：管理员

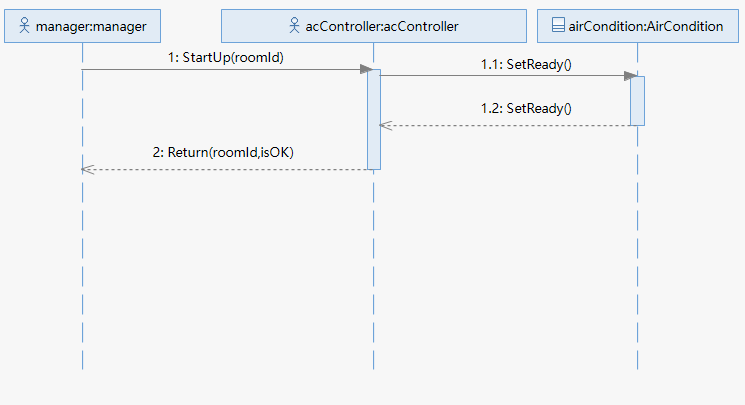
用例：运行空调

功能：管理员将从控机启动运行



->: StartUp(RoomId)

<-: Return(RoomId,isOK)



1. 管理员通过空调控制器启动空调服务

1.1 空调控制器设置对应的空调对象的状态属性为：就绪

1.2 空调对象状态设为就绪

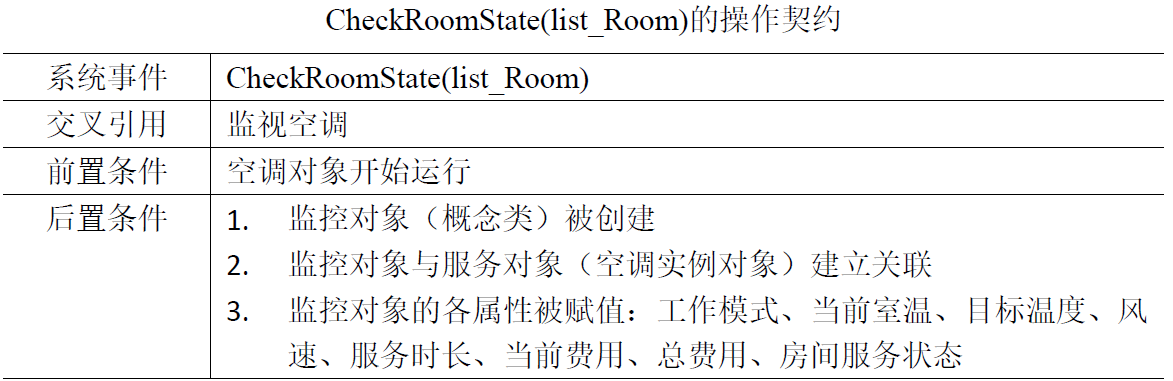
2. 空调启动成功

### 3.4.4 CheckRoomState检查空调状态

角色：管理员

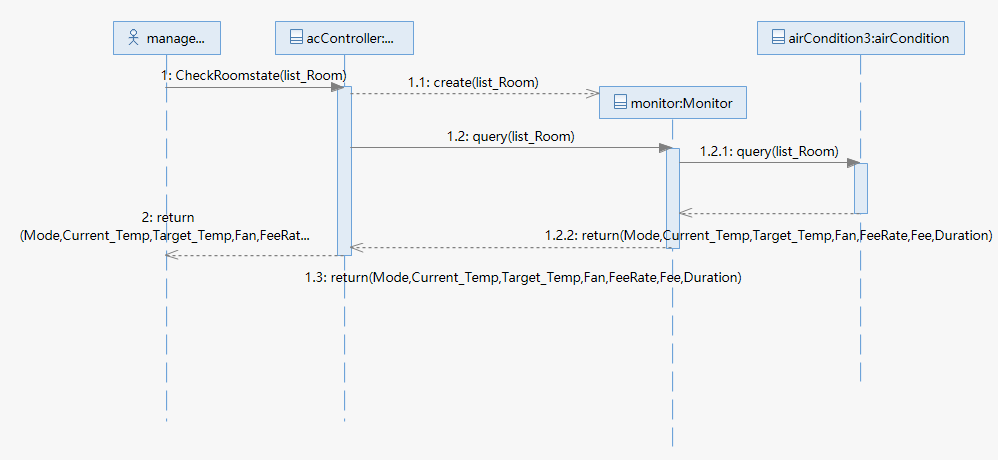
用例：监视空调

功能：获取各个从控机的状态参数进行监控



->: CheckRoomState(List\_Room)

<-: Return(Mode,Current\_Temp,Target\_Temp,Fan,Duration,Fee,Total\_Fee,state)



1. 管理员向contrller查询list\_room房间信息

1.1 Controller创建监视对象，并通过list\_Room与多个空调对象建立关联

1.2 Controller向监视对象查询当前运行参数

1.2.1 监视对象向空调对象查询当前运行参数

1.2.2 空调对象向监视对象返回当前运行参数

1.3 监视对象向Controller返回当前运行参数

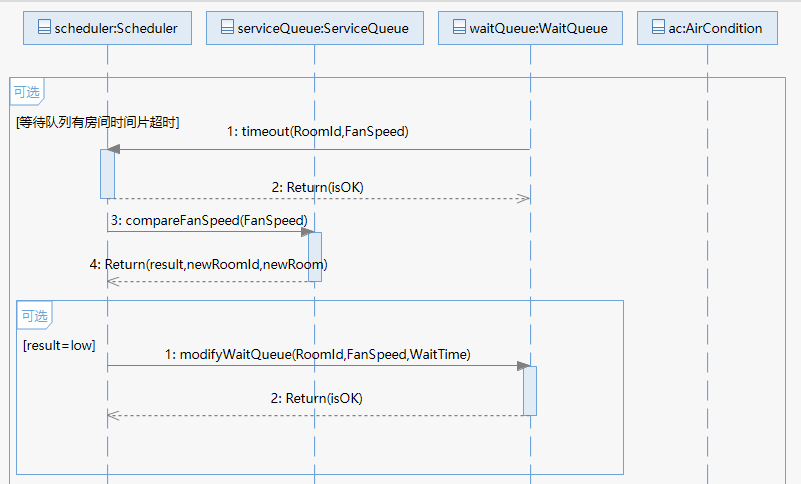
2 监视空调操作完成

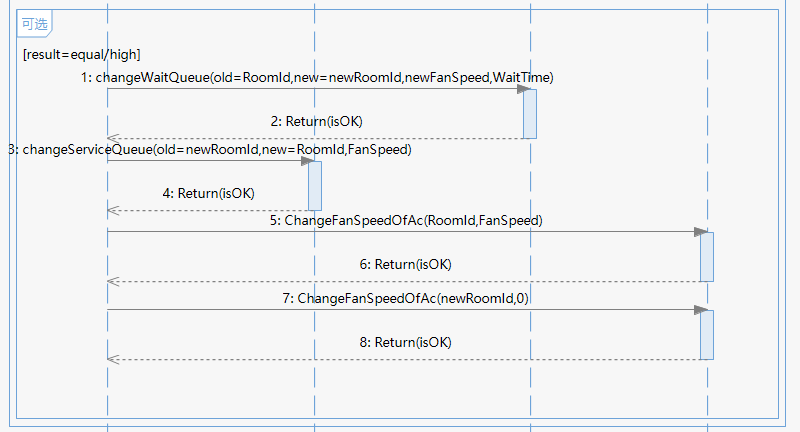
# 4 关于调度

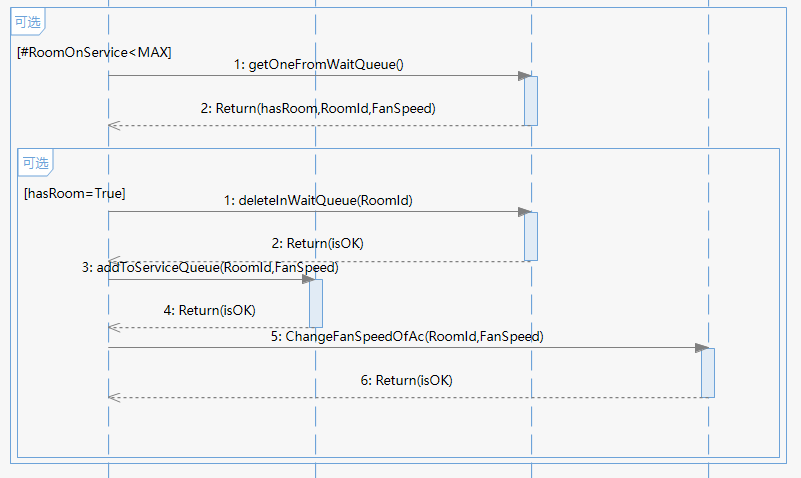
1）服务队列替换房间的规则：替换掉风速最小的房间；如果风速相同，替换服务时间最长的房间；

2）等待队列取出房间的规则：取出风速最大的房间；如果风速相同，取出时间片剩余最小的房间；

以下部分说明与角色无关的调度部分：







**当等待队列中有时间片超时：**

1 等待队列通知调度器有超时

3 将超时房间的风速与服务队列中的进行比较

[如果比服务队列中的风速低]

1 用modify方法刷新时间片

[如果存在至少一个比超时风速小的，选出最适合被替换的，称为“新房间”]

1 将等待队列中的超时房间替换为新房间及风速

3 将服务队列中的新房间替换为超时房间及风速

5 将超时房间的风速设为FanSpeed

7 将新房间的风速设为0

**当服务队列中房间数小于最大数值：**

1 从等待队列取一个新房间

[如果取到了一个房间]

1 在等待队列中删除该房间

3 将该房间加入服务队列

5 向空调对象发送启动该空调的消息

# 5 文档分工

于越：用户的调度部分，文档整合

王岚睿：管理员部分（共4个）

刘宇航：前台部分（共4个）

李文淇：经理部分和客户的后三个（共5个）

唐方柳：客户部分的前四个（共4个）