

**软件需求规格说明书**



**题目： 波普特酒店空调系统动态结构设计**

**班 级： 2017211309**

**组 号： 软件工程309-d**

**学 院： 计算机学院**

**2020年 5 月 22 日**

**文件修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **修订人** | **修订日期** | **修订内容** |
| 1.0 | 程年智 | 2020-05-19 | 创建、编写文件初稿 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 概述 5](#_Toc41067264)

[1.1约定 5](#_Toc41067265)

[1.2 预期读者和阅读建议 5](#_Toc41067266)

[1.3 产品范围 5](#_Toc41067267)

[1.4 参考文献 5](#_Toc41067268)

[2 项目背景描述 6](#_Toc41067269)

[2.1 运行环境 6](#_Toc41067270)

[2.2 设计和实现上的限制 7](#_Toc41067271)

[2.3 假定和依赖 7](#_Toc41067272)

[3 系统软件架构说明 8](#_Toc41067273)

[3.1 架构概述 8](#_Toc41067274)

[3.2 架构示意图 8](#_Toc41067275)

[4动态结构设计 9](#_Toc41067276)

[4.1 客户的系统事件 9](#_Toc41067277)

[4.1.1 RequestOn 顾客开机 10](#_Toc41067278)

[4.1.2 ChangeTargetTemp 房间修改空调温度 12](#_Toc41067279)

[4.1.3 ChangeFanSpeed 修改风速 13](#_Toc41067280)

[4.1.4 . RequestOff 顾客关闭空调 14](#_Toc41067281)

[4.1.5 sendFee 客户端代缴费用更新机制 15](#_Toc41067282)

[4.2 前台的系统事件 16](#_Toc41067283)

[4.2.1 CreateRDR 创建详单 17](#_Toc41067284)

[4.2.2 PrintRDR 打印详单 18](#_Toc41067285)

[4.2.3 CreateInvoice 创建账单 19](#_Toc41067286)

[4.2.4 PrintInvoice 打印详单 20](#_Toc41067287)

[4.3 管理员的系统事件 21](#_Toc41067288)

[4.3.1 PowerOn 打开空调 22](#_Toc41067289)

[4.3.2 SetPara() 设置启动参数 23](#_Toc41067290)

[4.3.3 *StartUp*  启动空调服务 24](#_Toc41067291)

[4.3.4 监视空调 25](#_Toc41067292)

[4.4 经理的系统事件 26](#_Toc41067293)

[4.4.1 queryReport 查看报表 27](#_Toc41067294)

[4.4.2 printReport 打印报表 28](#_Toc41067295)

[4.4.3 deleteReport 删除报表 28](#_Toc41067296)

[4.5 调度规则 29](#_Toc41067297)

[4.5.1 等待队列时间片耗尽 30](#_Toc41067298)

[4.5.2 服务队列服务对象服务完成或关机 33](#_Toc41067299)

[4.5.3 客户修改服务队列 35](#_Toc41067300)

[4.5.4 客户修改等待队列中的风速 38](#_Toc41067301)

[5 分工说明 42](#_Toc41067302)

# 1 概述

本系统描述的是位于帝都五环外的波普特（BUPT）廉价酒店的空调自动温控计费系统。

本系统面向四类用户，空调的使用者客户，酒店前台人员，空调的管理者和酒店经理。我们使用xx开发环境为这四类用户开发了简单易上手的操作面板，用户可以使用系统轻松的完成自己想要做的任务。

本系统安装在酒店房间，中央空调和前台与酒店经理房间，使用酒店网络进行控制。使用本系统成本较低，酒店房间安装一个操控面板，管理人员可以使用windows系统电脑进行管理，易于上手，通信使用酒店网络。同时我们将对系统提供定期维护与升级功能，提升系统的稳定性。

建设本系统，用于完成甲方对于空调管理的需求和对政府绿色环保经营的理念。相信如此便捷，实用的系统一定可以吸引更多年轻人的来往。

**本文档的目的是从设计者的角度阐述系统的动态架构，帮助甲方及使用者了解我方对系统架构的设计，相信阅读完这篇文档，可以让阅读者对系统的结构有更深刻的理解。**

## 1.1约定

本文档的描述内容是波普特酒店自动温控系统的动态结构设计。文档的功能需求列表参照双方共同认可的报价列表内容。文档采用基于UML建模语言的面向对象建模方式对功能需求进行描述。

文档中所有的UML图形均以 IBM RSA 8.0.3版本的建模工具进行绘制。

性能需求以文字和列表的方式具体给出。

## 1.2 预期读者和阅读建议

|  |  |  |
| --- | --- | --- |
| 编号 | **预期读者** | **阅读建议** |
| **1** | 客户 | 确认文档中给出的功能需求描述 |
| **2** | 开发方 | 熟悉并掌握项目的各项功能要求 |

## 1.3 产品范围

本系统适用于波普特酒店的中央空调系统。面向四类用户：客户，前台服务员，空调管理员，酒店经理。功能主要有用户控制空调和各种账单报表的打印。

## 1.4 参考文献

《波普特酒店空调系统建设调研报告》

《波普特酒店空调系统设计领域模型》

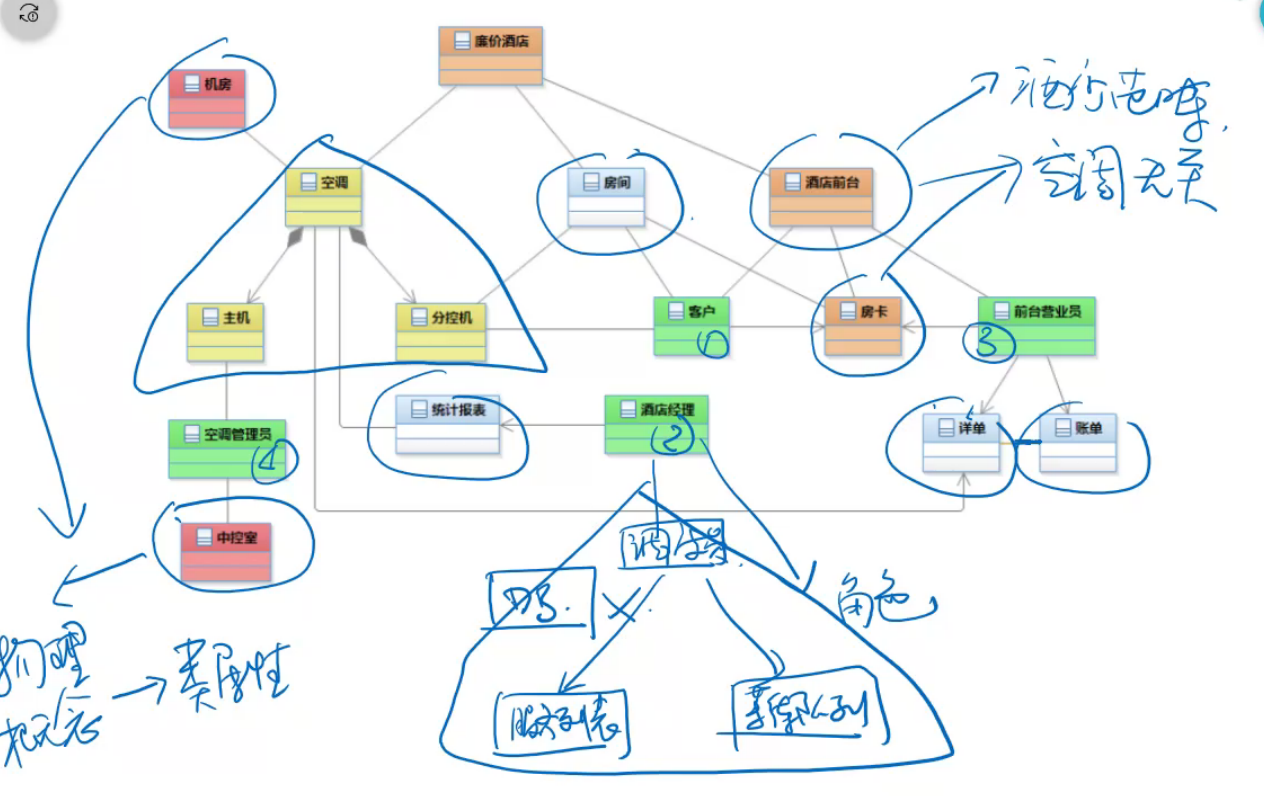
《波普特酒店空调系统设计软件需求规格说明书》

# 2 项目背景描述

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。

更多信息可以参考《波普特酒店空调系统建设调研报告》。

系统架构可以参考领域模型中的类图，可以有效说明系统的结构特征。



## 2.1 运行环境

|  |  |  |
| --- | --- | --- |
| 编号 | **名称** | **运行环境** |
| **1** | 应用服务器 | *Ubuntu16及以上* |
| **2** | Web服务器 | *Tomcat 6.0.26* |
| **3** | 数据库 | *Oracle 11g r2* |
| **4** | 数据存储 | *Oracle RAC* |
| **5** | 并行计算平台 | *Hadoop 0.20.2 / hive 0.11.0* |
| **6** | 客户端 | *IE8及以上浏览器，Firefox, Google chrome* |

## 2.2 设计和实现上的限制

## 2.3 假定和依赖

# 3 系统软件架构说明

## 3.1 架构概述

层次化的设计模型是面向对象方法基于软件体系结构风格的一种方案选择。使用层次化的设计模型可以让系统结构清晰，功能的封装更完善，代码的实现也更简单。

本系统采用了5层软件架构，分别是用户界面层、控制器层、应用层、持久化层和数据库，分别承担数据库中的不同职责。

用户界面层：用户进行交互的部分，代表客户端，包含应用程序中用户界面部分的代码，是直接与使用者交互的部分。

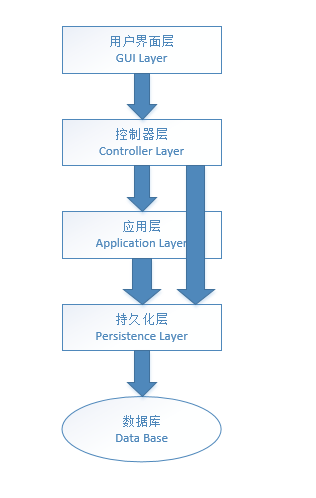
控制器层：负责接收用户的请求或者系统事件的部分，再将这些请求或者系统事件发送给下面几层。控制器层的作用主要是将所有对请求或系统事件的响应封装在一起，再转发给其他部分。本系统中控制器层主要有用户使用空调的用例控制器调度对象和另外三个角色公用的外观控制器空调对象。

应用层:系统的主要部分，系统中的建立的对象和相应的逻辑处理都主要存储在这一层，应用层是实现功能的层次。

持久化层：系统中需要长期保存的数据存储在这个层次。某些需要经常读写或者需要持久化的数据如果都存储在数据库中，读写数据库时将会浪费很多时间，所以设立了持久化层管理（增删改查）经过业务逻辑对象处理后的需要持久保存的数据，又能与业务逻辑的功能相分离保持其独立性，又能与数据库保持同步。

数据库： 存储数据的关系型数据库，有增删改查等基本操作。

## 3.2 架构示意图



# 4动态结构设计

## 4.1 客户的系统事件

操作契约：



时序图中类的定义说明：

|  |  |
| --- | --- |
| 类名 | 说明 |
| Customer | 客户，进行空调使用 |
| Scheduler | 调度对象，客户使用空调用例控制器 |
| ServiceObj | 服务对象，提供空调服务功能 |
| queue<ServiceObj> | 服务队列，服务对象组成的队列 |
| RequstObj | 等待对象，记录客户使用空调请求 |
| queue<RequestObj> | 等待队列，等待对象组成的队列 |
| DBFacade | 数据库外观类，负责维护领域类和负责其持久化的对应Bean类之间的对应关系 |
| roomBean | 房间信息持久化类，负责房间信息持久化 |

### 4.1.1 RequestOn 顾客开机

RequestOn(RoomID, CurrentRoomTemp)

🡨🡪 Return(defaultFanSpeed, defaultMode, defaultTargetTemp, defaultFeeRate, defaultFee)

·传入参数：房间ID，当前房间温度。

·返回值：默认风速，默认风速模式，默认目标温度，默认费率，默认费用。

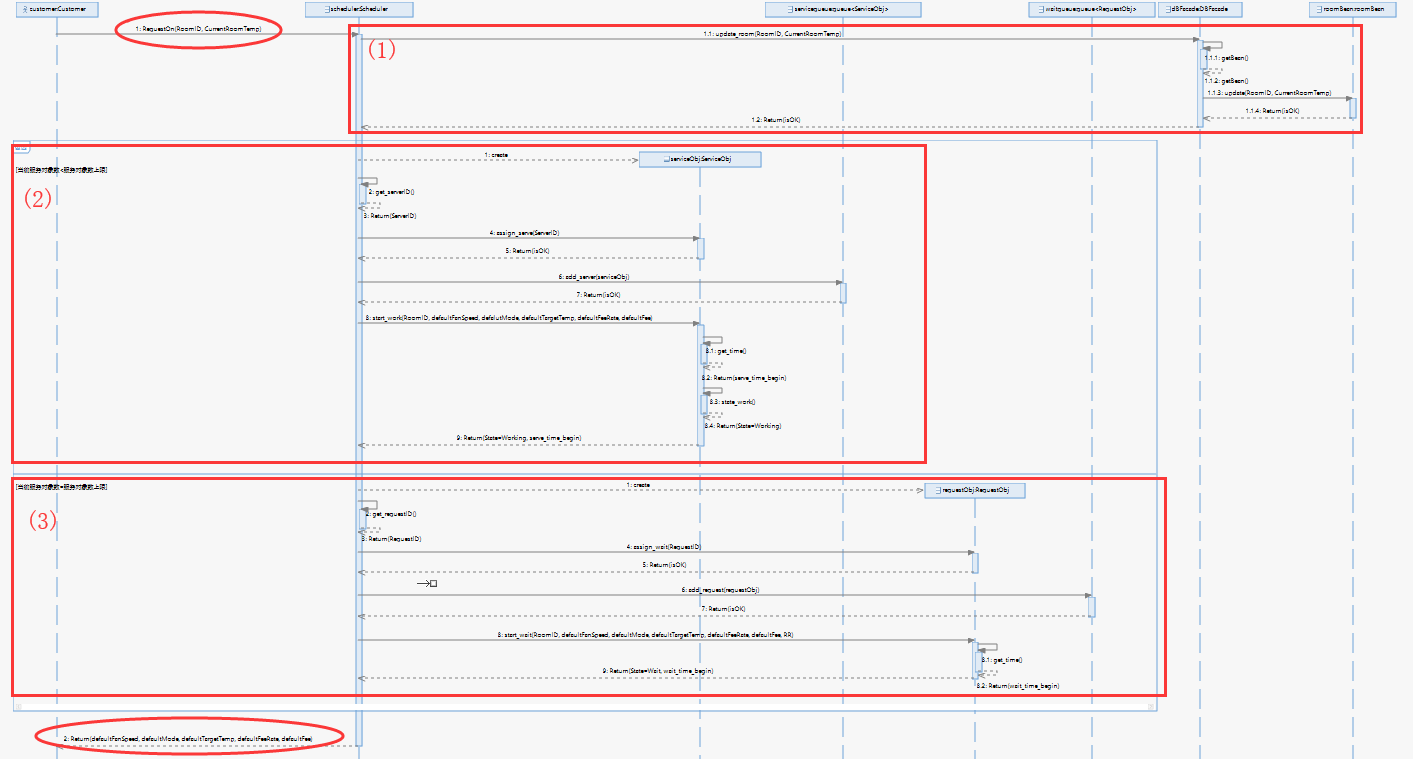
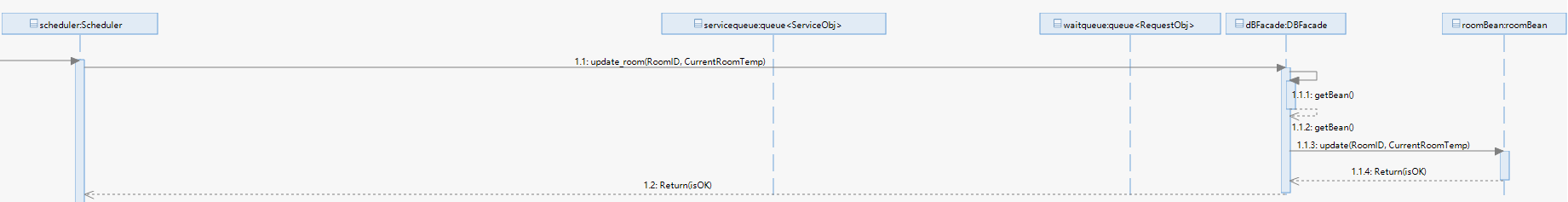


Figure 1 图片比较模糊，下面有3张分解的图片

为什么客户开机就会被放入服务队列或者调度队列，因为系统有默认的初始风速和初始温度，所有房间一开始优先级是相同的，所以都是使用的FIFO的时间片调度方式。等到客户修改风速（即优先级）时，优先级调度才会启动，这是我们小组的设计。

⑴修改数据库文件中的房间信息：

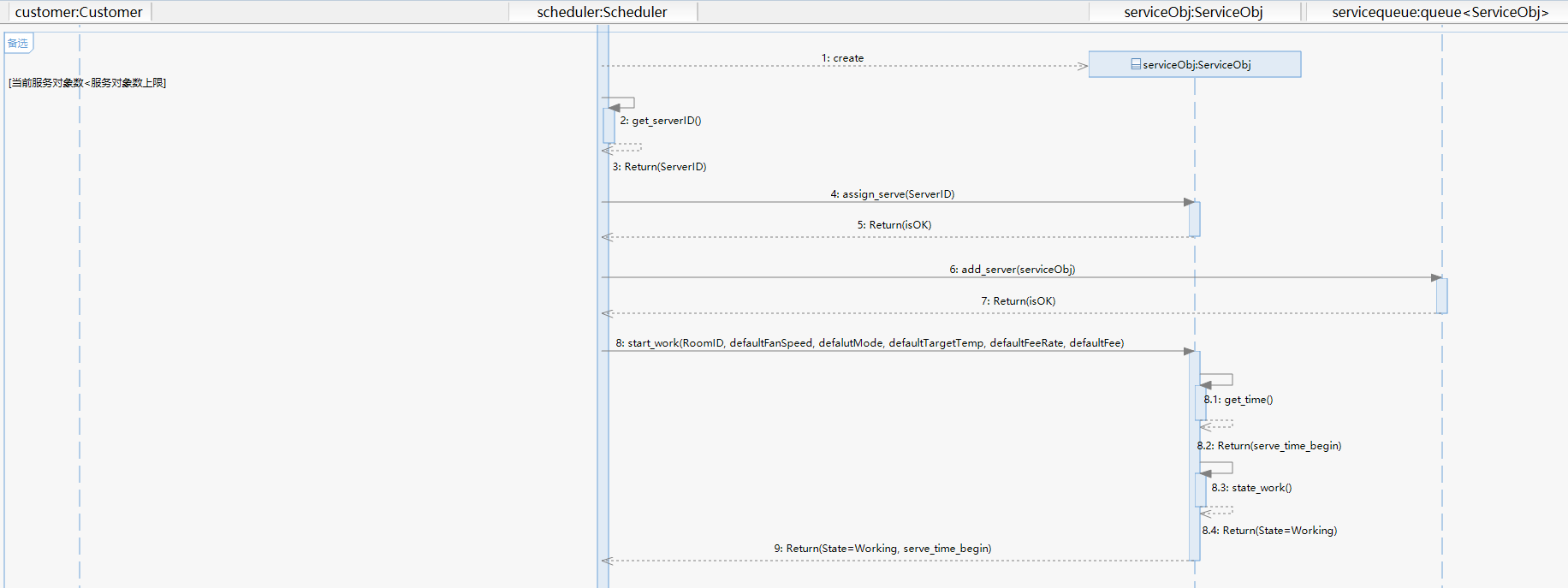


1、调度对象Scheduler向持久化层DBFacade对象发送更新消息，调用update\_room操作；

2、update\_room操作调用DBFacade对象的getBean()获取负责对Room对象进行持久化的对象roomBean；

3、update\_room操作调用roomBean对象的update操作，将相应房间对应的温度持久化到存储介质。

⑵当前服务对象数<服务对象数上限：



1、调度对象Scheduler创建服务对象ServiceObj实例serviceObj；

2、Scheduler获取应为serviceObj分配的ID；

3、Scheduler为serviceObj绑定该ID；

4、Scheduler将新创建的服务对象实例serviceObj加入服务队列queue<SerivceObj>；

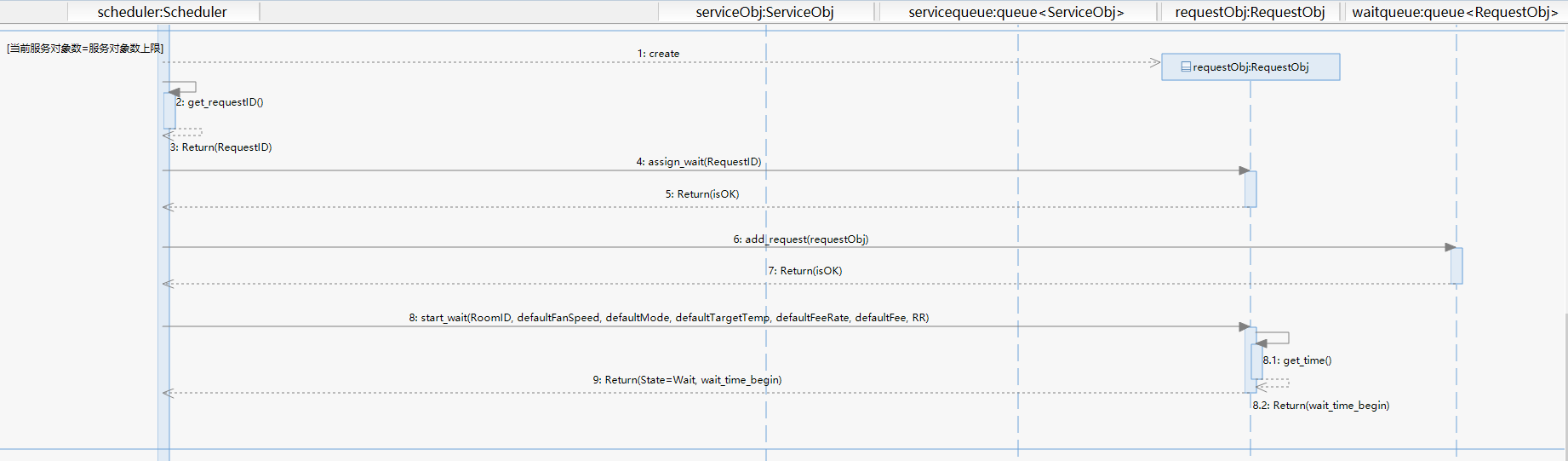
5、启动服务，Scheduler为serviceObj传入服务房间ID、默认的服务参数（风速、模式、温度）、默认的费率等信息；

5.1、serviceObj获取当前时间，为服务开始时间serve\_time\_begin；

5.2、serviceObj调用state\_work方法，正式开始服务，将服务状态修改为“正在工作（Working）”；

5.3、返回值为serviceObj的服务状态（Working）和服务开始时间（serve\_time\_begin）。

⑶当前服务对象数=服务对象数上限



1、调度对象Scheduler创建等待对象RequestObj实例requestObj；

2、Scheduler获取应为requestObj分配的ID；

3、Scheduler为requestObj绑定该ID；

4、Scheduler将新创建的等待对象实例requestObj加入等待队列queue<RequestObj>；

5、开始等待，Scheduler为requestObj传入服务房间ID、默认的服务参数（风速、模式、温度）、默认的费率、初始时间片片长（RR）等信息；

5.1、requestObj获取当前时间，为等待开始时间wait\_time\_begin；

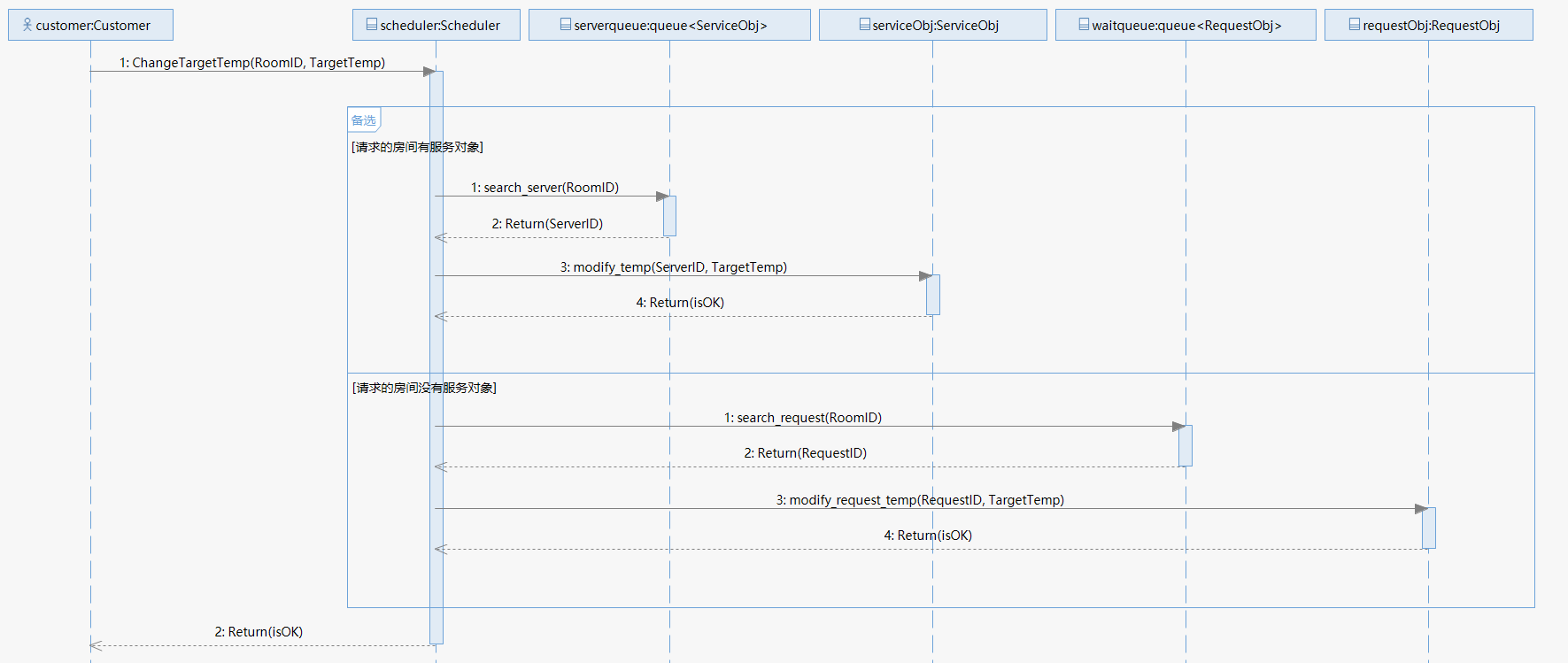
5.2、返回值为等待状态（Wait）和等待开始时间（wait\_time\_begin）。

### 4.1.2 ChangeTargetTemp 房间修改空调温度

ChangeTargetTemp(RoomID, TargetTemp) 🡨🡪 Return(isOK)

·传入参数：房间ID，目标温度。

·返回值：修改成功标识。



1、请求的房间有服务对象

①调度对象Scheduler根据房间ID查询对应的服务对象ID（ServerID）；

②Scheduler修改请求的房间对应的服务对象serviceObj服务参数，修改其目标温度为TargetTemp。

2、请求的房间没有服务对象

①调度对象Scheduler根据房间ID查询对应的等待对象ID（RequestID）；

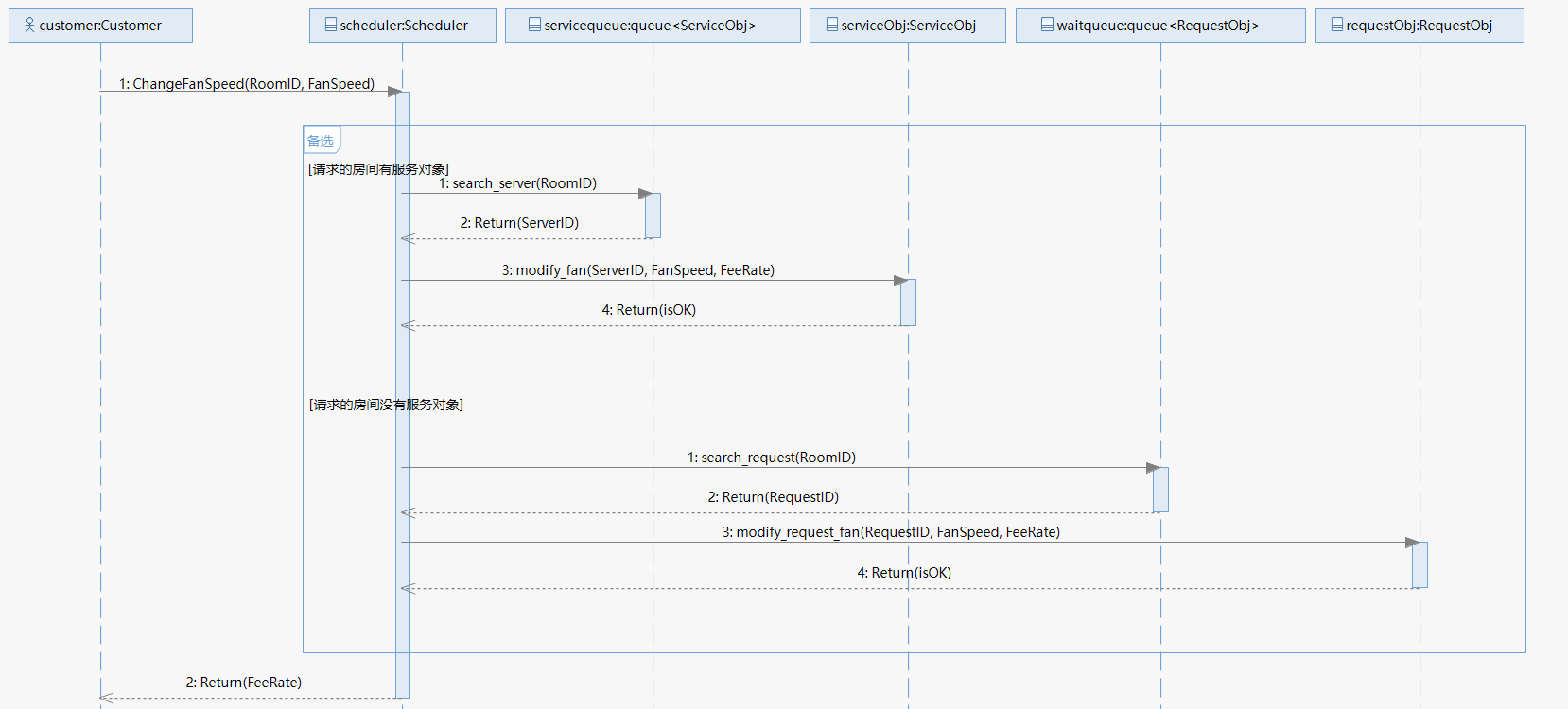
②Scheduler修改请求的房间对应的等待对象requestObj参数，修改其目标温度为TargetTemp。

### 4.1.3 ChangeFanSpeed 修改风速

ChangeFanSpeed(RoomID, FanSpeed) 🡨🡪 Return(FeeRate)

·传入参数：房间ID，目标风速。

·返回值：该风速对应的费率



1、请求的房间有服务对象

①调度对象Scheduler根据房间ID查询对应的服务对象ID（ServerID）；

②Scheduler修改请求的房间对应的服务对象serviceObj服务参数，修改其风速为FanSpeed；并向serviceObj传入该风速对应的费率FeeRate。

2、请求的房间没有服务对象

①调度对象Scheduler根据房间ID查询对应的等待对象ID（RequestID）；

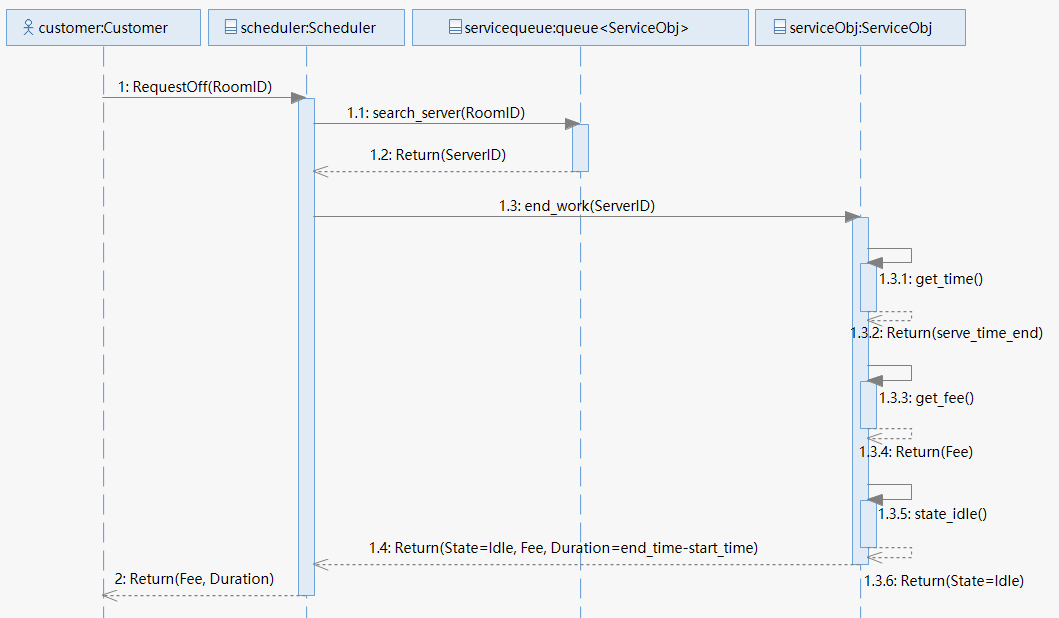
②Scheduler修改请求的房间对应的等待对象requestObj参数，修改其风速为FanSpeed、对应的费率为FeeRate。

### 4.1.4 . RequestOff 顾客关闭空调

RequestOff(RoomID) 🡨🡪 Return(Fee, Duration)

·传入参数：房间ID。

·返回值：本次服务的费用，本次服务的持续时长。（本次服务：从为房间指定本服务对象提供服务开始，到当前时刻服务结束）



1、调度对象Scheduler根据房间ID查询对应的服务对象ID（ServerID）；

2、Scheduler向请求的房间对应的服务对象serviceObj发送消息，使停止工作；

·①serviceObj获取当前时间，即为服务结束时间serve\_time\_end；

·②serviceObj调用get\_fee方法，获得本次服务的总费用Fee；

·③serviceObj调用state\_idle方法，停止服务，将服务状态修改为“空闲（Idle）”；

·返回值为serviceObj的服务状态（Idle）、本次服务总费用（Fee）和本次服务时长（Duration）。

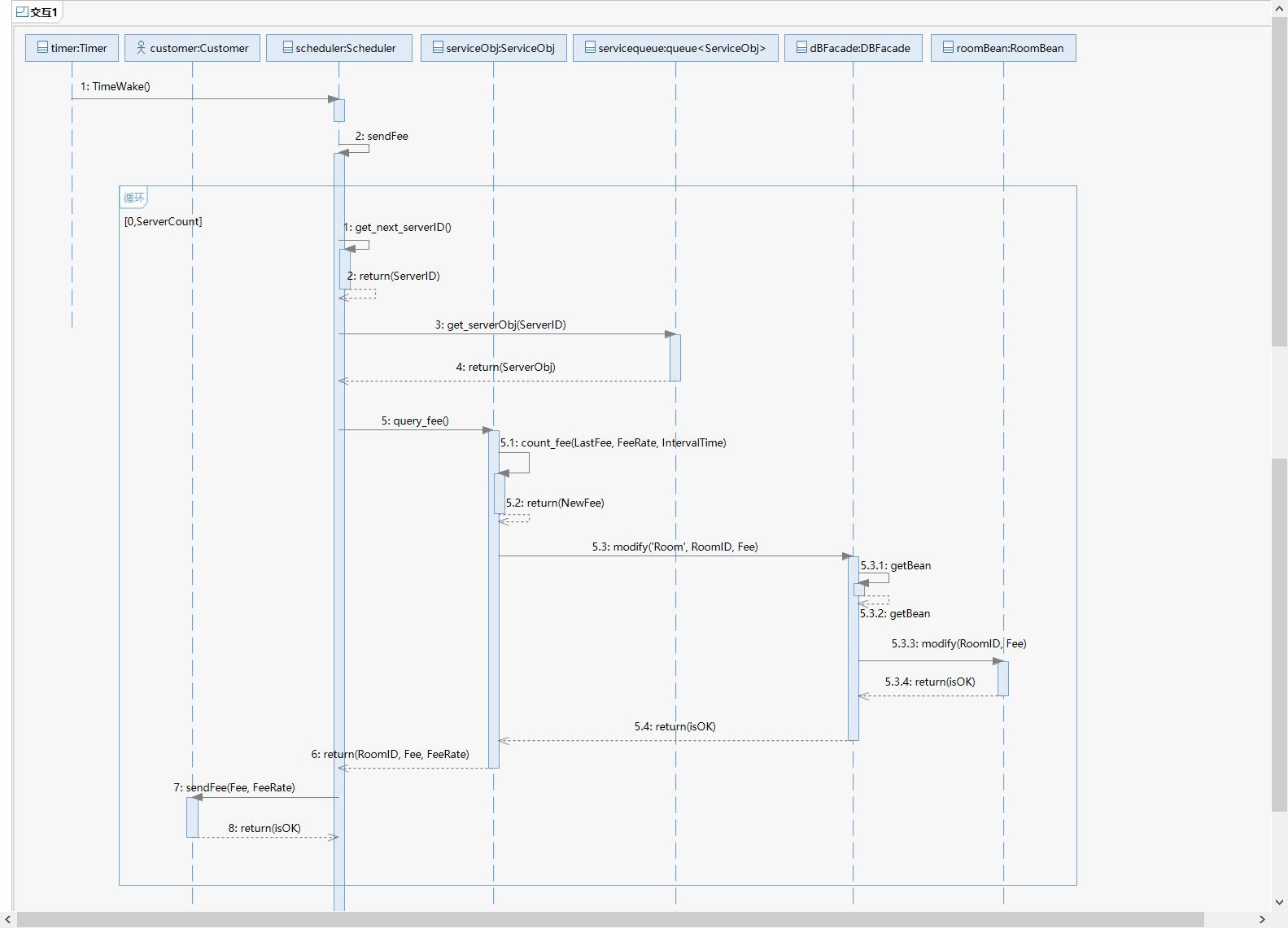
### 4.1.5 sendFee 客户端代缴费用更新机制

**角色**：客户

**用例**：使用空调

**函数**：sendFee(Fee, FeeRate)

**功能**：由服务器端主动发送，每分钟更新一次费用



**步骤：**

1. Timer向Scheduler发送定时信息（1分钟）。函数:TimeWake()
2. Scheduler收到定时信息后，为正在服务的房间更新费用。函数:sendFee()

循环遍历每个服务房间：

1. 获取当前服务对象；通过Scheduler中的get\_next\_serverID()，获取在队列中的位置，再通过向服务队列发送的get\_serverObj(ServerID)，获取服务对象。
2. 向该服务对象询问房间号、最新费用、费率。函数:query\_fee()
3. 由服务对象根据之前的费用，费率，间隔时间，计算出最新费用。函数:count\_fee(LastFee, FeeRate, IntervalTime)
4. 向DBFacade发送modify(‘Room’, RoomID, Fee)修改数据库中相关数据，DBFacade收到后根据getBean()将参数’Room’（领域类名）关联到持久化对象RoomBean上，并发送modify(RoomID, Fee)进行修改
5. 根据房间号，更新对应房间的最新费用与费率。函数:sendFee(Fee, FeeRate)

## 4.2 前台的系统事件

操作契约：



时序图中类的定义说明：

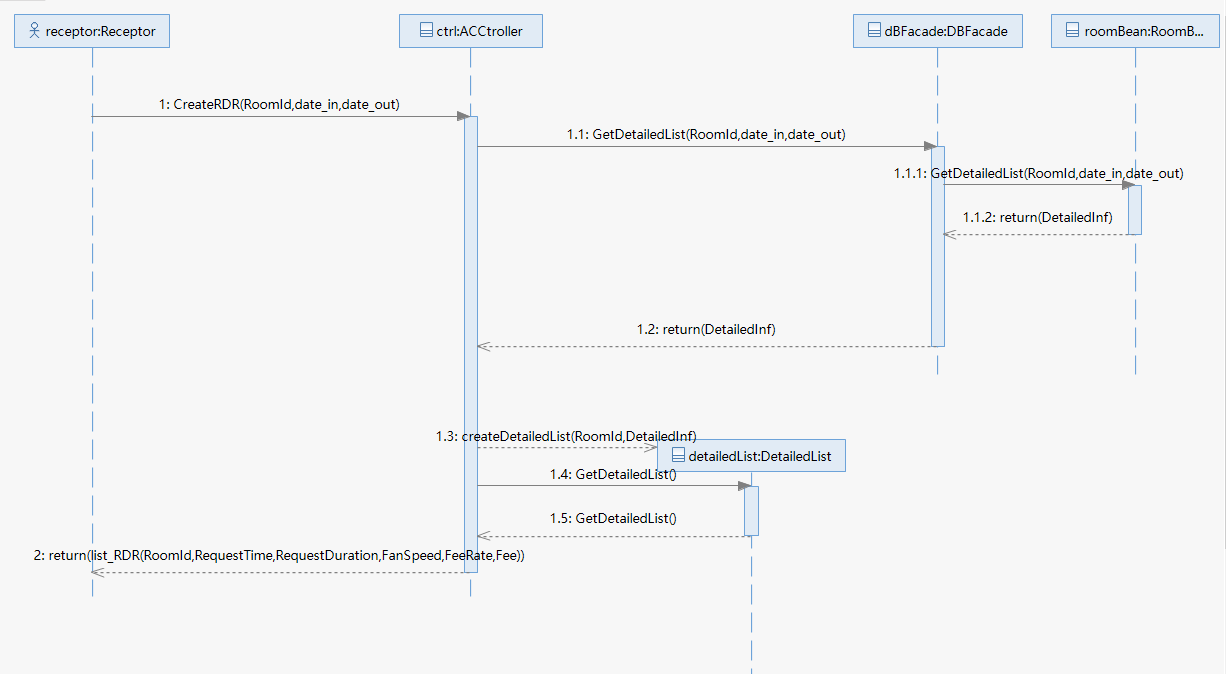
|  |  |
| --- | --- |
| 类名 | 说明 |
| Receptor | 前台接待员 |
| ACCtroller | 控制器 |
| DBFacade | 持久化操作数据库 |
| RoomBean | 房间数据持久化操作对象 |
| DetailedList | 详单 |
| Invoice | Invoice |

### 4.2.1 CreateRDR 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



1. 前台接待员的请求被控制器接收
   1. 控制器将请求转发给持久化操作数据库
      1. 房间数据持久化操作对象通过RoomId，date\_in，date\_out三个数据在数据库中查询数据
      2. 返回生成详单所需要的数据
   2. 返回生成详单所需要的数据
   3. 创建详单
   4. 获取详单内容
   5. 返回详单内容

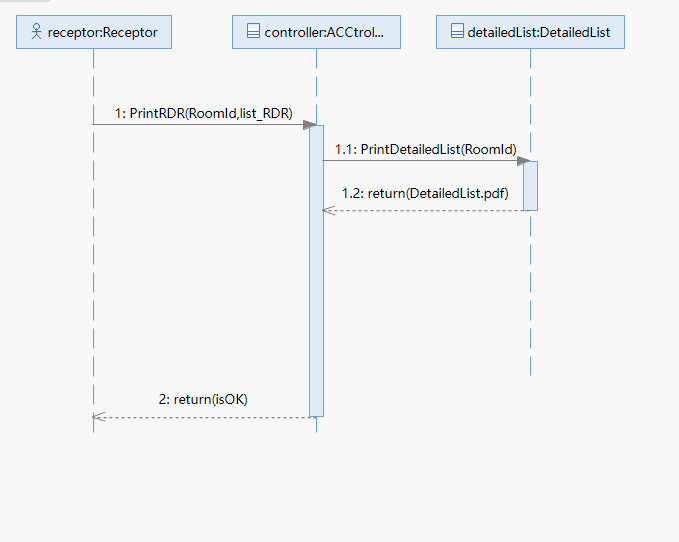
2.返回详单内容

### 4.2.2 PrintRDR 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



1. 前台接待员的请求被控制器所接收
   1. 根据房间id打印详单
   2. 返回pdf格式的详单

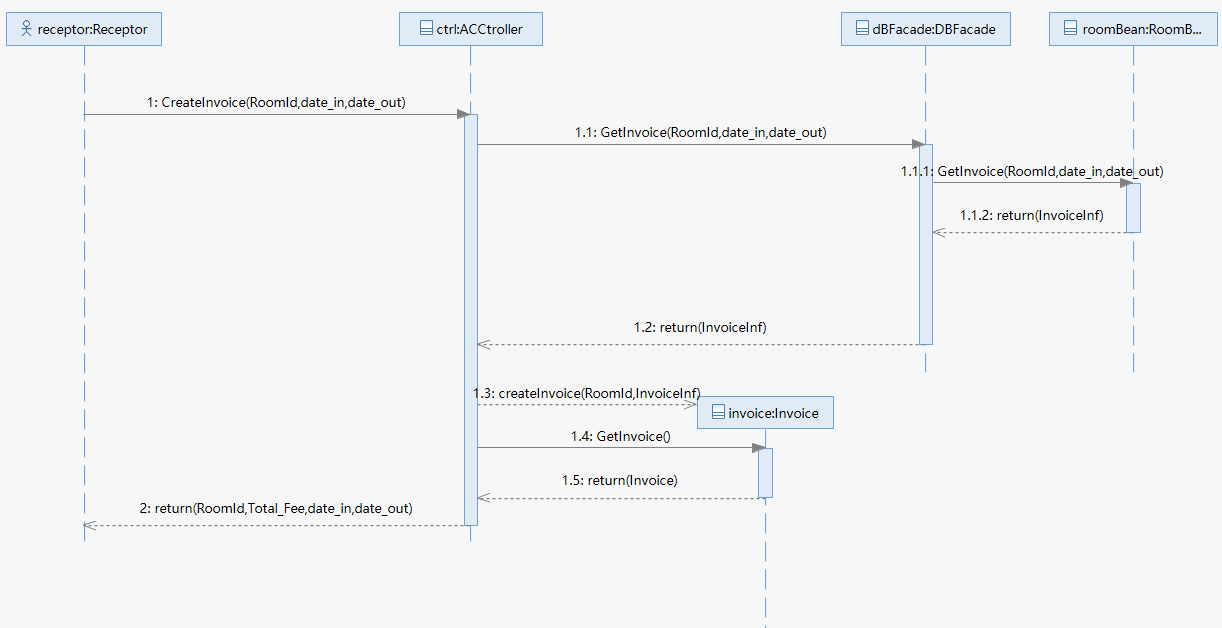
2.返回是否打印成功

### 4.2.3 CreateInvoice 创建账单

角色：前台

用例：出账单

功能：结账时前台接待员创建账单



1. 前台接待员的请求被控制器接收

1.1控制器将请求转发给持久化操作数据库

1.1.1房间数据持久化操作对象通过RoomId，date\_in，date\_out三个数据在数据库中查询数据

1.1.2返回生成账单所需要的数据

1.2 返回生成账单所需要的数据

1.3 创建账单

1.4 获取账单内容

1.5 返回账单内容

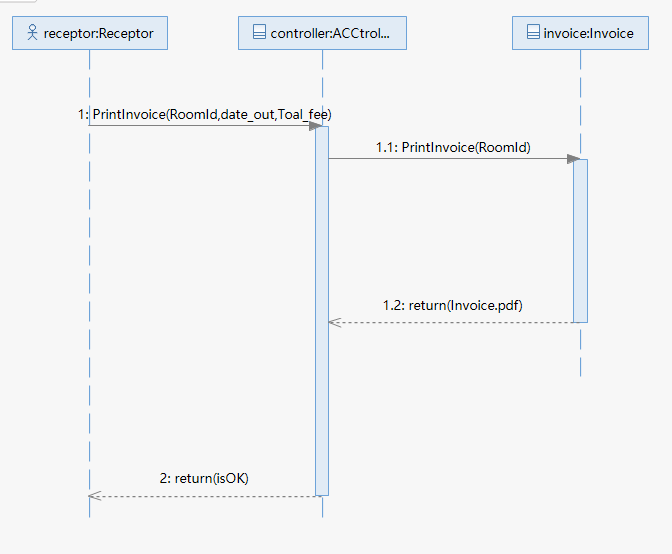
2.返回账单内容

### 4.2.4 PrintInvoice 打印详单

角色：前台

用例：出账单

功能：结账时前台接待员打印账单



1.前台接待员的请求被控制器所接收

1.1根据房间id打印账单

1.2返回pdf格式的账单

2.返回是否打印成功

## 4.3 管理员的系统事件

操作契约：

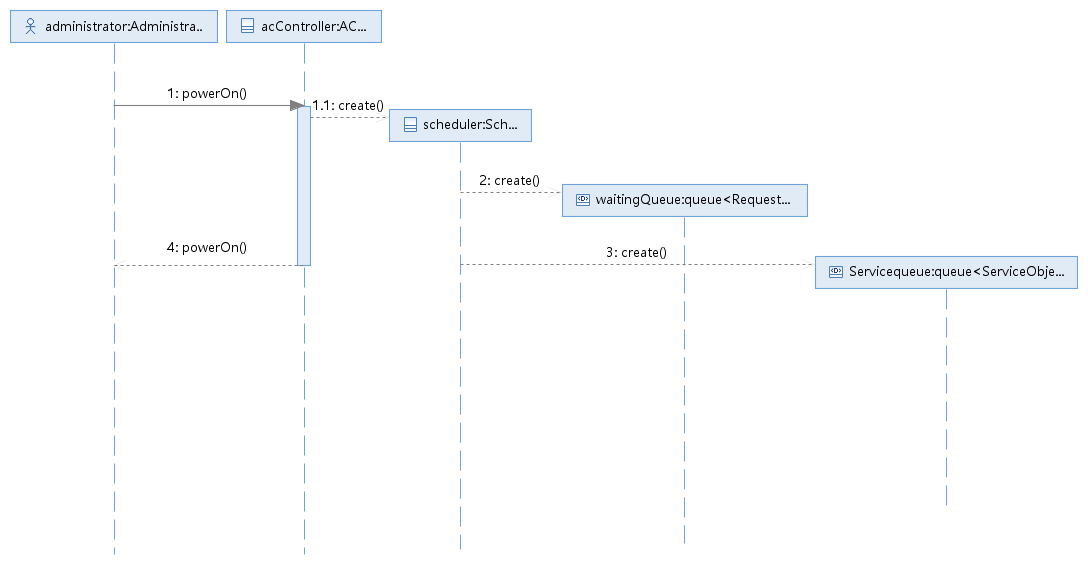


类说明：

|  |  |
| --- | --- |
| 类名称 | 说明 |
| Administrator | 管理员 |
| ACController | 外观控制器 |
| Scheduler | 调度对象 |
| DBFacade | 持久化操作数据库 |
| RoomBean | 房间信息的持久化对象 |
| Serviecequeue | 服务队列 |
| waitingQueue | 等待队列 |

### 4.3.1 PowerOn 打开空调

管理员打开空调电源，进入设置模式。



1. 管理员将打开空调的指令发给空调控制器（打开电源）

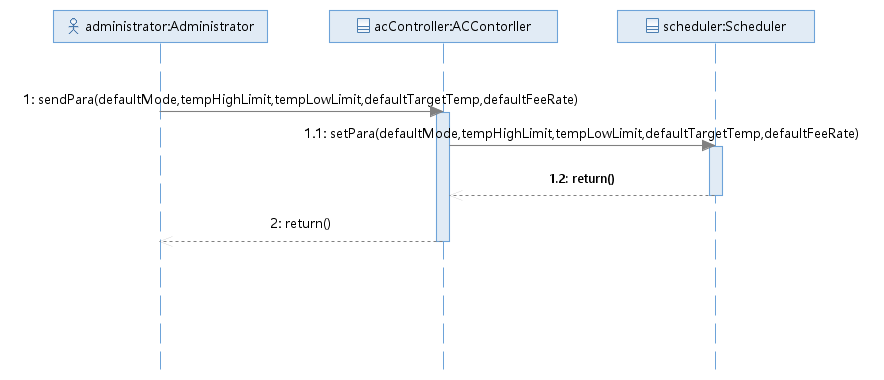
1.1. 空调控制器创建调度对象，调度对象生成并初始化，再由调度对象创建等待队列和服务队列

2. 调度对象创建等待队列

3. 调度对象创建服务队列

4. PowerOn()完成

### 4.3.2 SetPara() 设置启动参数



1. 管理员在界面里输入初始化服务对象的各项参数给空调控制器，其中：

defualtMode为服务对象创建时默认的模式；

tempHighLimit为温度最高限制；

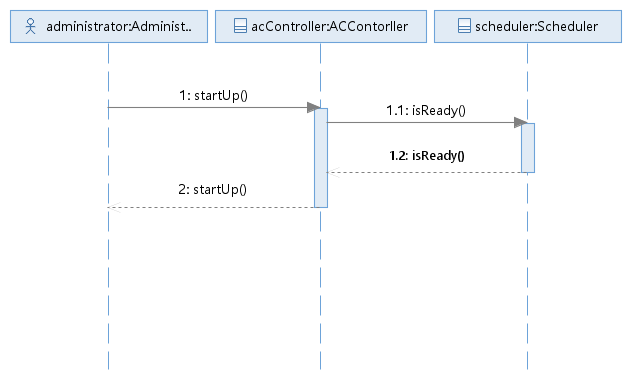
tempLowLimit为温度最低限制；

defaultTargetTemp为默认目标温度；

defaultFeeRate为默认费率。

1.1. 参数传给调度对象，调度对象创建服务对象的时候使用这些参数初始化服务对象，参数具体意义同上

### 4.3.3 *StartUp* 启动空调服务

**

1. 管理员通过空调控制器启动空调服务

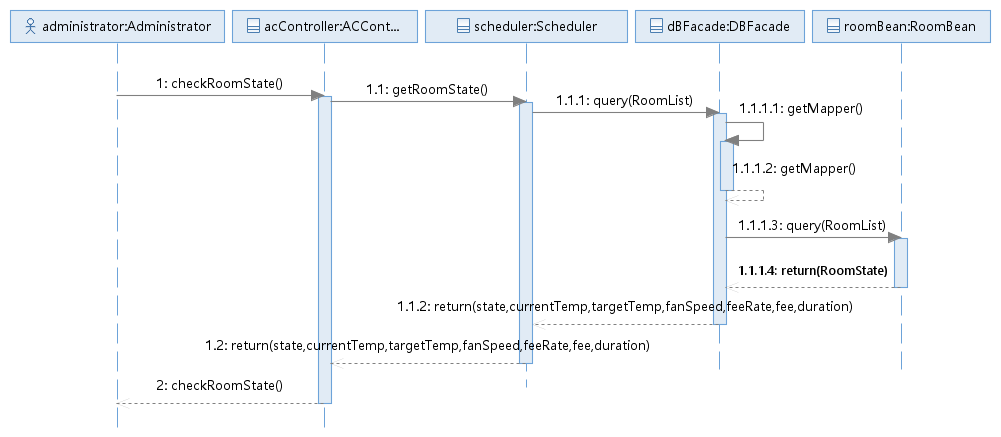
1.1. 空调控制器把调度对象的状态设置为：就绪

1.2. 调度对象状态已设为就绪

2. 空调成功启动

### 4.3.4 监视空调

管理员通过GUI监视各个房间的状态。



1. 管理员通过空调控制器监视各房间空调状态

1.1. 空调控制器调用调度对象获得房间状态

1.1.1. 调度对象向DBFacade查询当前所有房间的状态

1.1.1.1. DBFacade找到对应的Mapper用来获取RoomBean对象的信息

1.1.1.3. DBFacade查询房间的状态

1.1.1.4. DBFacade得到各个房间的状态

1.1.2. 调度对象得到各个房间的状态：

state：状态信息（是否开机，是否正在被服务，是否被挂起等）；

currentTemp：当前室温；

targetTemp：目标温度；

fanSpeed：风速；

feeRate：费率；

fee：费用；

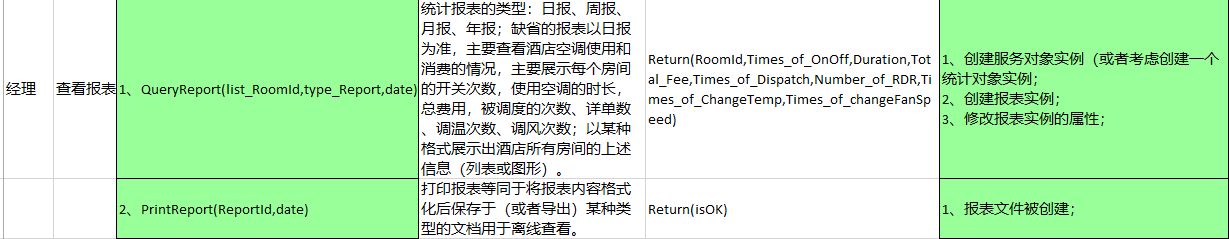
duration：服务时长。

1.2. 调度对象把各个房间的状态返回给空调控制器

2. 管理员得到各个房间的状态信息

## 4.4 经理的系统事件

操作契约：



类说明

|  |  |
| --- | --- |
| **Manager** | 酒店经理 |
| **ACCtroller** | 用例控制器 |
| **DBFacade** | 维护领域类与持久化对象类的关系 |
| **RoomBean** | 房间数据的持久化对象 |

### 4.4.1 queryReport 查看报表

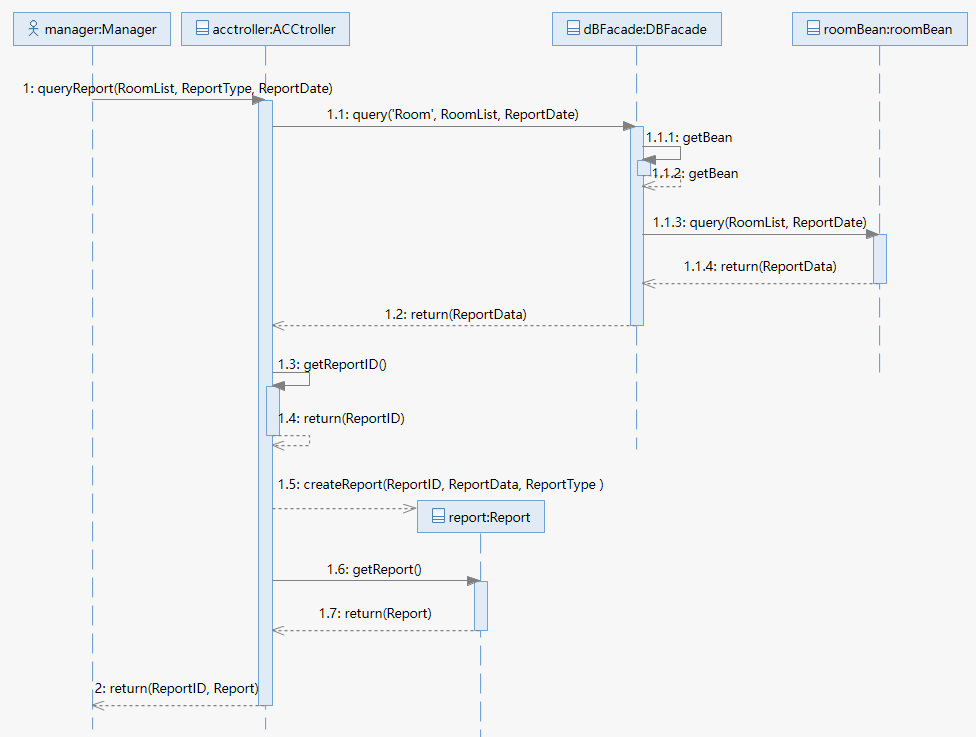
**角色**：经理

**用例**：查看报表

**函数**：queryReport(RoomList, ReportType, ReportDate)

return(Report)

**功能**：统计报表的类型：日报、周报、月报、年报；缺省的报表以日报为准，主要查看酒店空调使用和消费的情况，主要展示每个房间的开关次数，使用空调的时长，总费用，被调度的次数、详单数、调温次数、调风次数；以某种格式展示出酒店所有房间的上述信息（列表或图形）



步骤：

1. Manager向ACCtroller请求获取某种类型、某个时间段的一些房间的报表。

函数: queryReport(RoomList, ReportType, ReportDate)

1. ACCtroller收到请求后
2. 向DBFacade发送query('Room', RoomList, ReportDate)询问数据库中相关数据，DBFacade收到后根据getBean()将参数’Room’（领域类名）关联到持久化对象RoomBean上，并发送query(RoomList, ReportDate)询问相关数据。
3. 得到报表所需数据后，利用getReportID()分配新的报表ID，并根据报表数据以及要生成的报表类型创建新的报表对象createReport(ReportID, ReportData, ReportType )。向报表对象发送getReport()以获取生成的报表。

3） ACCtroller返回报表信息

### 4.4.2 printReport 打印报表

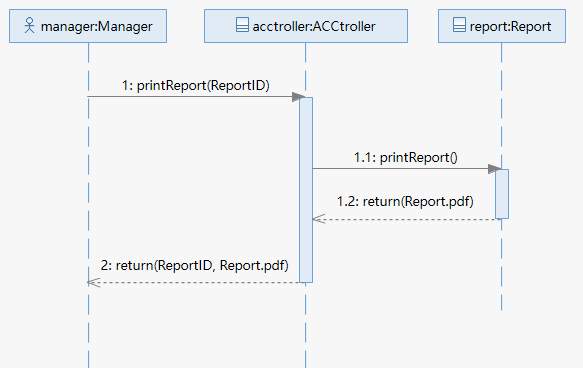
**角色：**经理

**用例**：打印报表

**函数**：printReport(ReportId)

return(ReportId, Report.pdf)

**功能**：在查看报表的基础上选择，打印报表等同于将报表内容格式化后保存于某种类型（此处为pdf格式）的文档用于离线查看。



### 4.4.3 deleteReport 删除报表

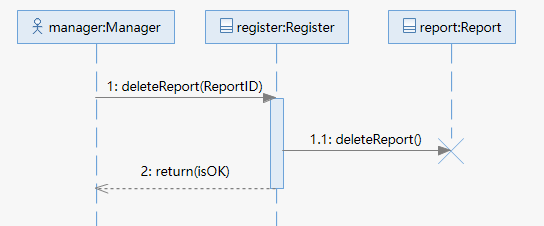
**角色**：经理

**用例**：查看报表

**函数**：deleteReport(ReportId)

return(isOK)

**功能**：退出报表页面的后续工作。



## 4.5 调度规则

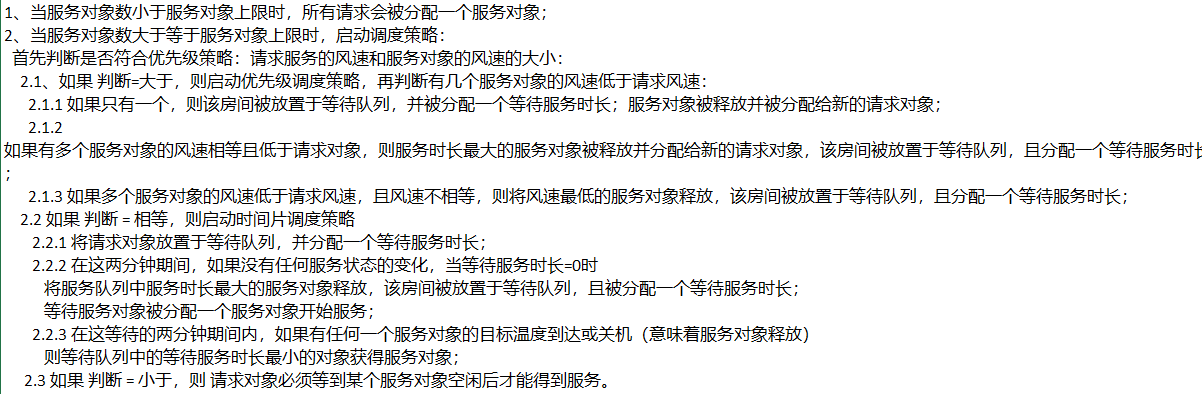
类的定义

|  |  |
| --- | --- |
| 类名 | 类的含义 |
| Customer | 客户 |
| Scheduler | 调度对象，客户的用例控制器 |
| RequestObject | 请求对象（假定每个房间只能有一个请求对象） |
| ServiceObject | 服务对象（系统内只有3个服务对象，由前三个使用空调的客户创建） |
| Servicequeue | 服务队列，存储服务对象的列表，有少许几种封装好的方法。 |
| Waitqueue | 等待队列，存储请求对象的列表，有少许几种封装好的方法。 |
| listservice | 客户修改等待队列中的对象的优先级时，从服务队列中获取的等待调度的服务对象列表，有少许几种封装好的方法。 |
| dbFacade | 维护领域类与持久化对象类的关系 |
| roomBean | 持久化的房间对象，存储着有关房间的信息，如当前温度，风速，代缴费用，主键是RoomID。 |

调度规则说明：

1、调度规则主要有两种，优先级调度和时间片调度，优先级调度先于时间片调度，优先级取决于客户的房间空调的设置风速，风速越高优先级越高，优先级越高的越优先服务。优先级相同的房间采用时间片调度，时间片调度首先遵循先来先服务，如果有空闲的服务对象就先被服务，否则被分配一个时间片置于等待队列中，等待空闲再被服务。

2、具体调度规则如下：



3、基于如上描述，首先用户开机时所有客户都使用的默认风速，优先级相同，所以所有客户都使用时间片调度。优先级调度使用应激式相应，当两个队列中的优先级改变时（即用户修改风速），使用优先级调度的策略改变队列中的服务状态。

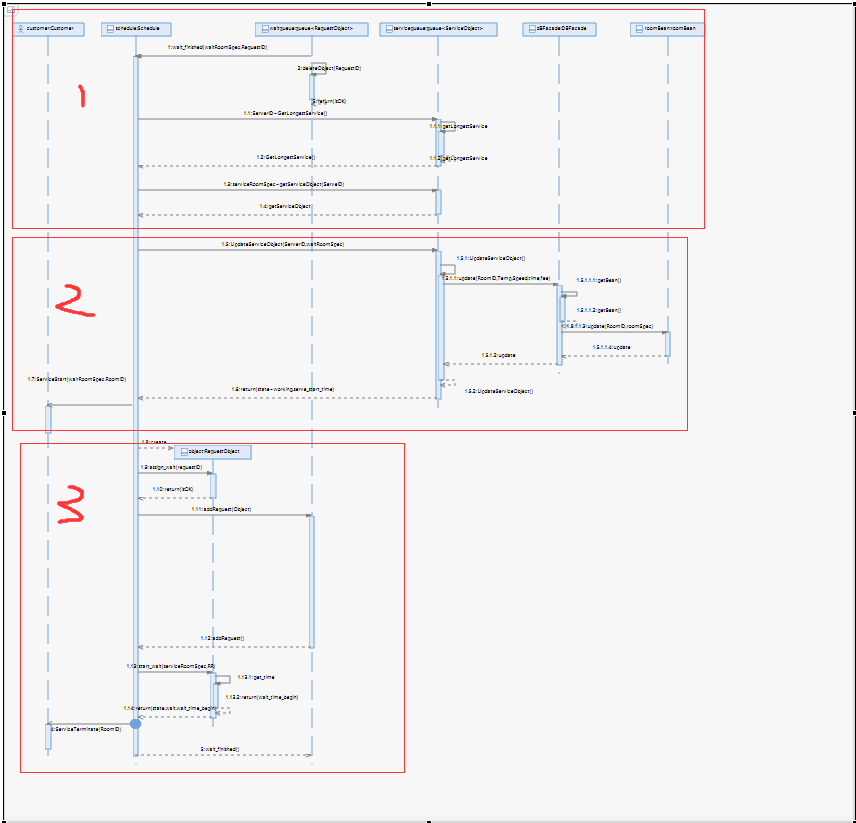
4、使用4个时序图来说明4种队列发送改变的情况，分别是：服务完成或客户关机，时间片耗尽，服务队列中优先级改变，等待队列优先级改变。

### 4.5.1 等待队列时间片耗尽

前置条件：等待队列中的某个请求对象的时间片耗尽，有等待队列向调度对象发送服务请求。

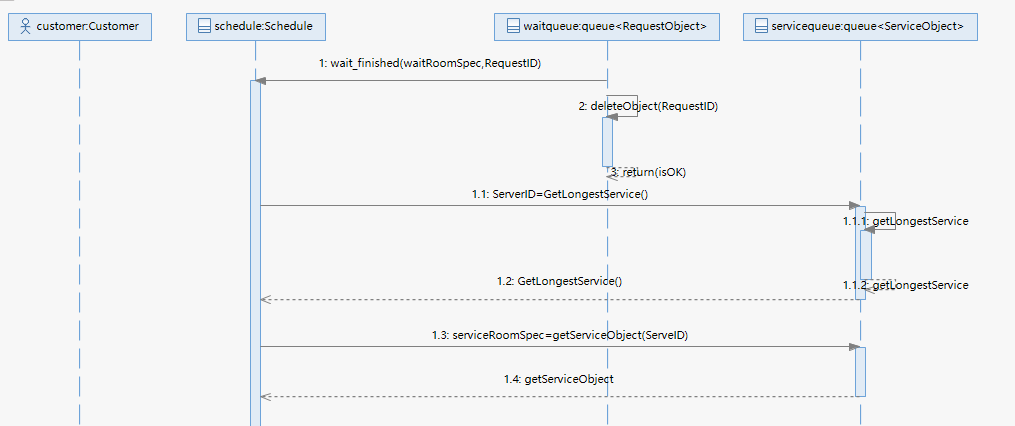
调度策略简要说明：将服务队列中某个服务时间最长的对象释放，分配一个全新的时间片进入等待队列，时间片耗尽的请求对象获得一个服务对象并且开始服务。

时序图：



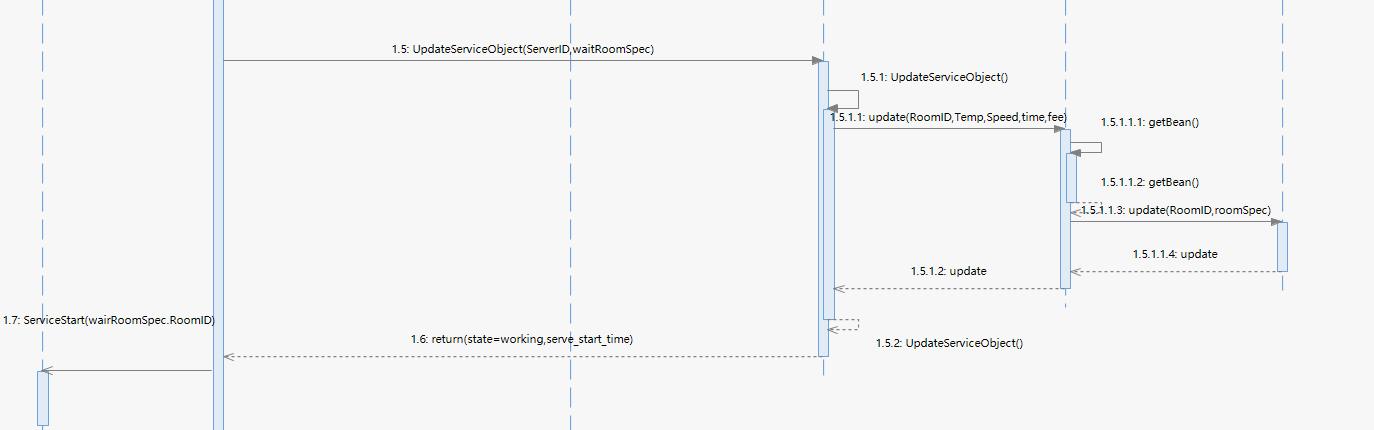
下面分别给出1 2 3 的详细图。

1、获取房间信息



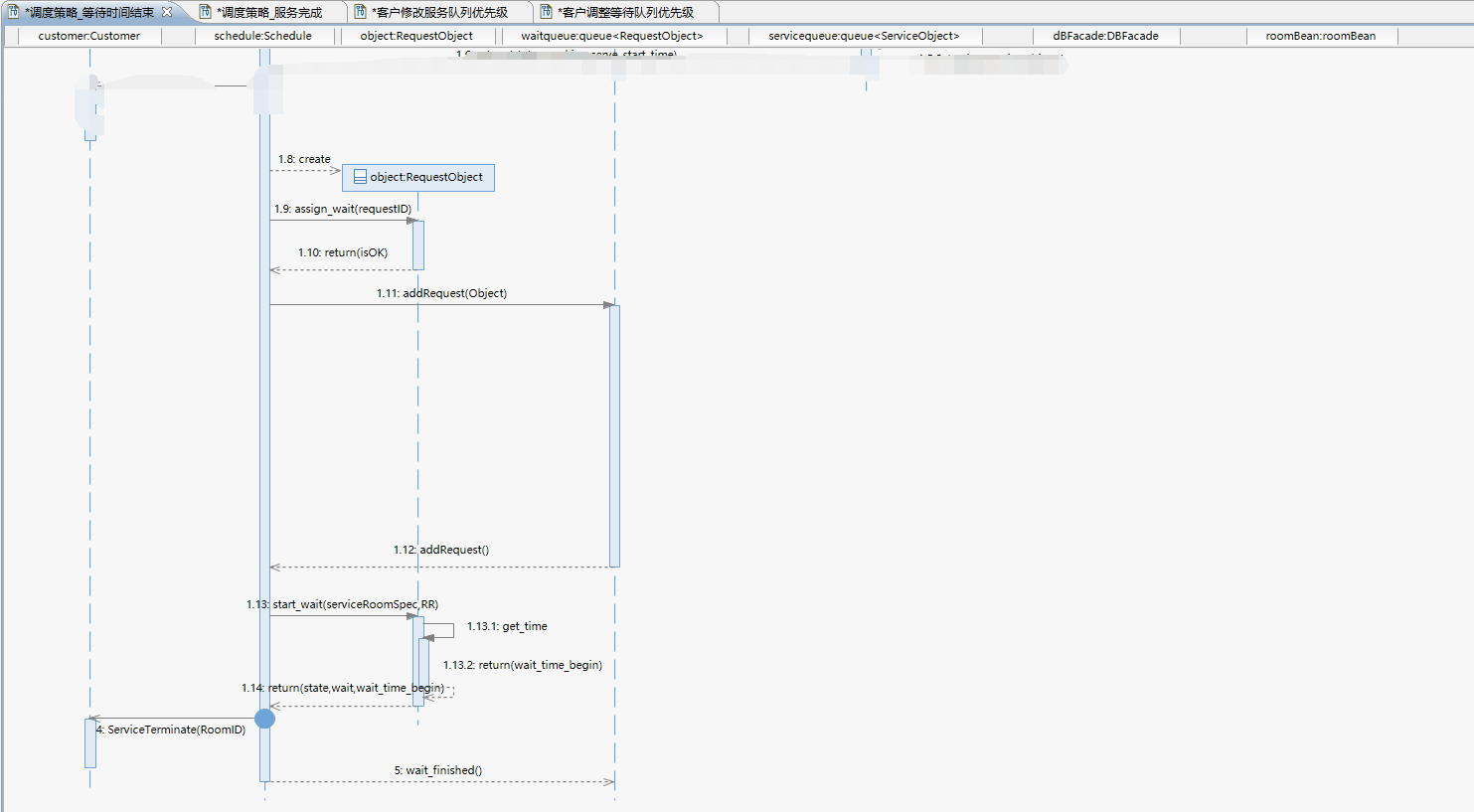
1. 等待队列发送wait\_finished给调度对象，说明拥有RequestID的的请求对象等待结束，并且附带该对象的信息waitroomSpec（包括RoomID,TargetSpeed,TargetTemp等信息）。
2. 等待队列使用deleteObject删除该请求对象。
3. 调度对象使用GetLongestService获取服务队列中服务时长最大的服务对象的serverID。
4. 调度对象根据serverID使用getServiceObject获取服务队列里的服务对象的数据。

2、将时间完的请求对象至于服务队列中。



1. 调用updateServiceObject,参数是希望更新的服务对象的ID和请求对象的信息。
2. 服务队列内部调用update方法，对服务对象进行更新，实际上就是为请求对象分配了一个服务对象。
3. 在更新前，首先使用dbfacade的update的方法对持久化层进行更新，将停止服务的服务对象的数据存储数据库中。
4. 然后从请求对象的信息中获取RoomID,向该房间的客户发送信息ServiceStart,说明该房间开始服务。

3、将停止服务的服务对象至于等待队列



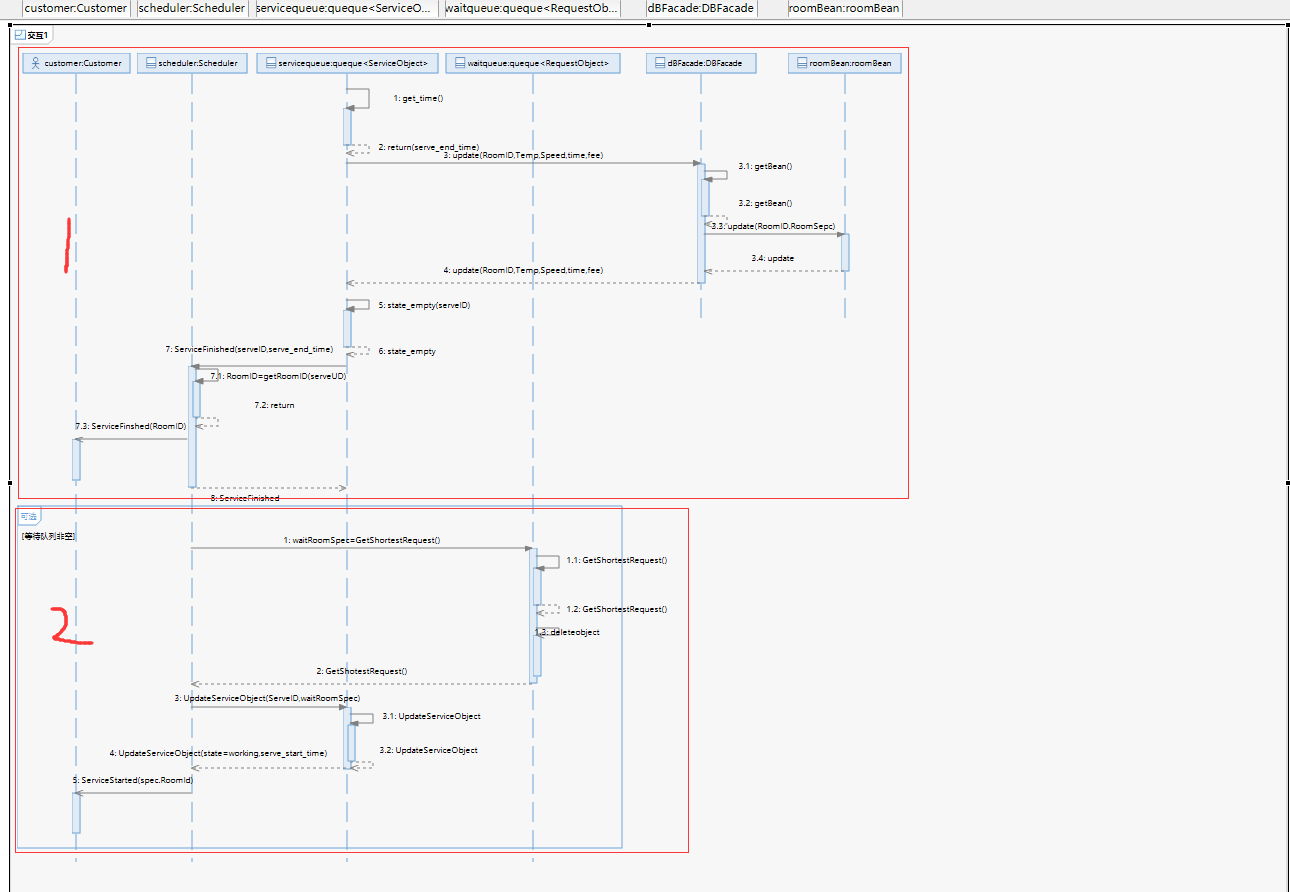
1. 创建请求对象object，并且使用RequestID标志其在队列中的位置。
2. 将object置于等待队列中。
3. 使用房间信息和时间片为object赋值。
4. 根据房间信息的RoomID找到该房间并且给客户回一个ServiceTerminate，说明服务以及终止，进入等待状态，并且附带当前最后的费用信息。

### 4.5.2 服务队列服务对象服务完成或关机

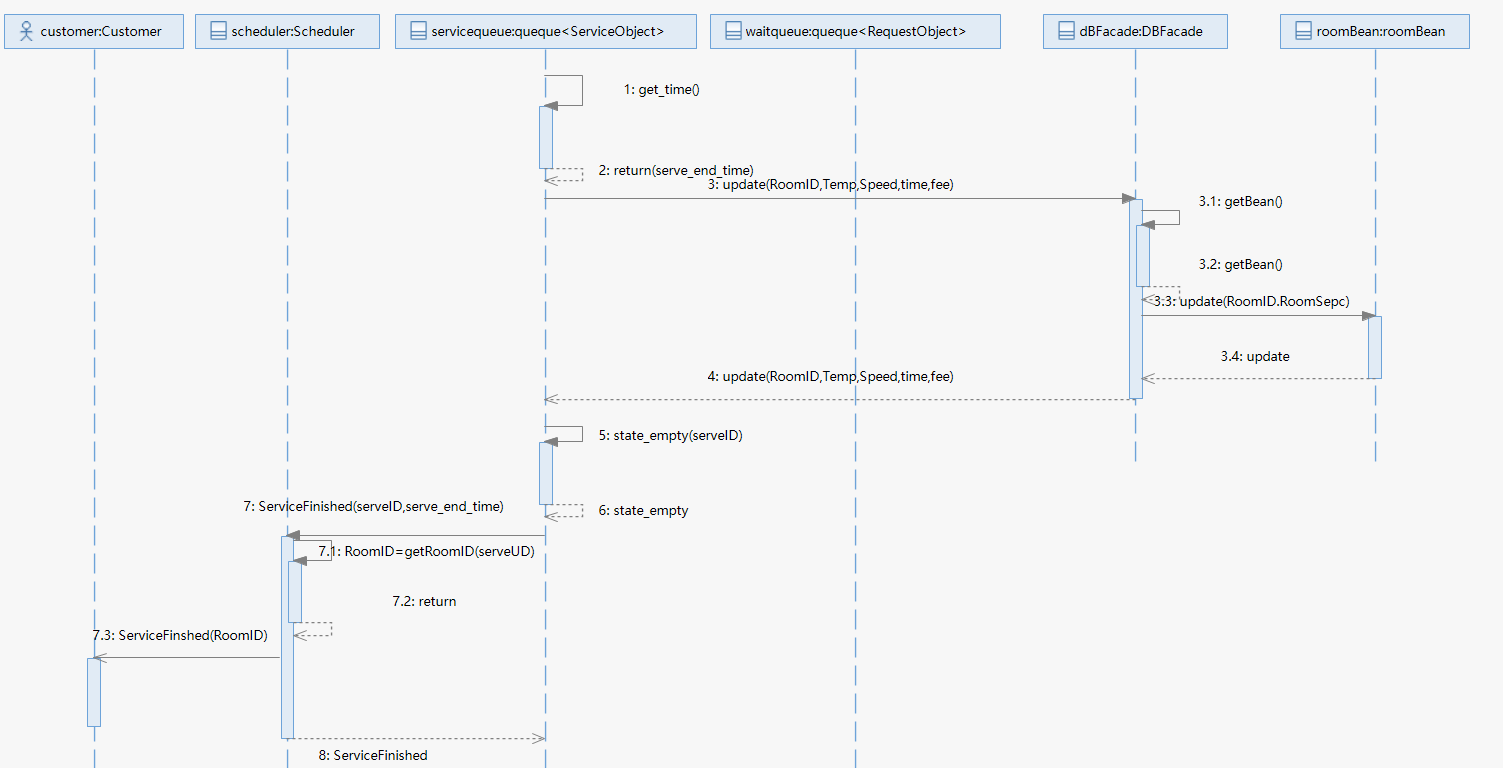
前置条件：服务队列中的某个服务队列服务完成服务或客户关机，服务队列向调度对象发出请求说明服务完成。

调度策略：从等待队列中获取时间片最短的请求对象，服务对象首先释放再分配个这个请求对象。

时序图：

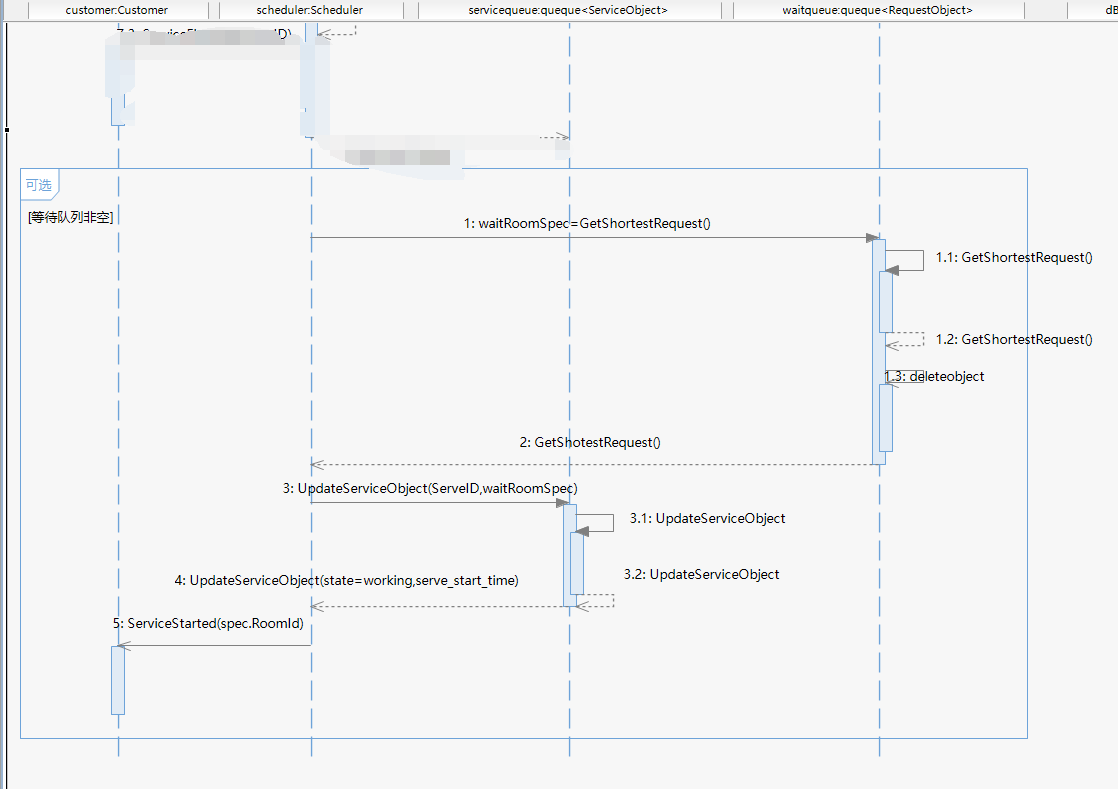


1、 服务完成或客户关机的处理



1. 服务队列使用get\_time获取服务结束时间，并且使用update将服务结束的对象的信息存入数据库中。
2. 设置服务对象状态为非繁忙，并且向调度对象发送结束服务的对象的serverID,费用，结束时间。
3. 根据serverID得到RoomID,向该房间的用户说明服务介绍，并携带最后一次的费用信息。

2、当等待队列非空时，给等待队列中时间片最短的请求对象分配服务对象。



若等待队列非空：

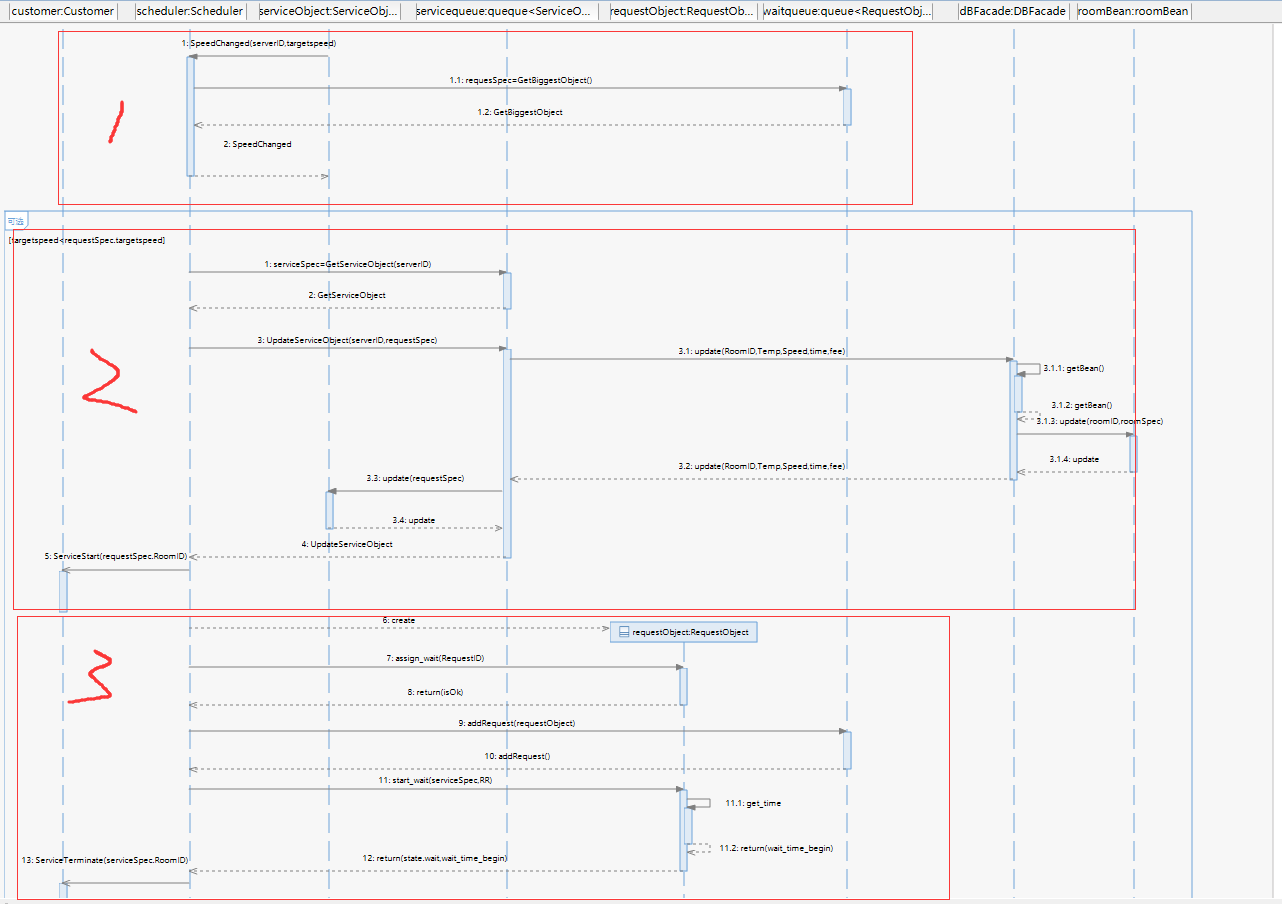
1. 通过调用getShortestObject方法获取时间片最短的请求对象的信息spec。
2. 等待队列调用deleteObject删除这个对象。
3. 调用updateServiceObject,参数是希望更新的服务对象的ID和请求对象的信息。
4. 服务队列内部调用update方法，对服务对象进行更新，实际上就是为请求对象分配了一个服务对象。
5. 在更新前，首先使用dbfacade的update的方法对持久化层进行更新，将停止服务的服务对象的数据存储数据库中。
6. 然后从请求对象的信息中获取RoomID,向该房间的客户发送信息ServiceStart,说明该房间开始服务。

### 4.5.3 客户修改服务队列

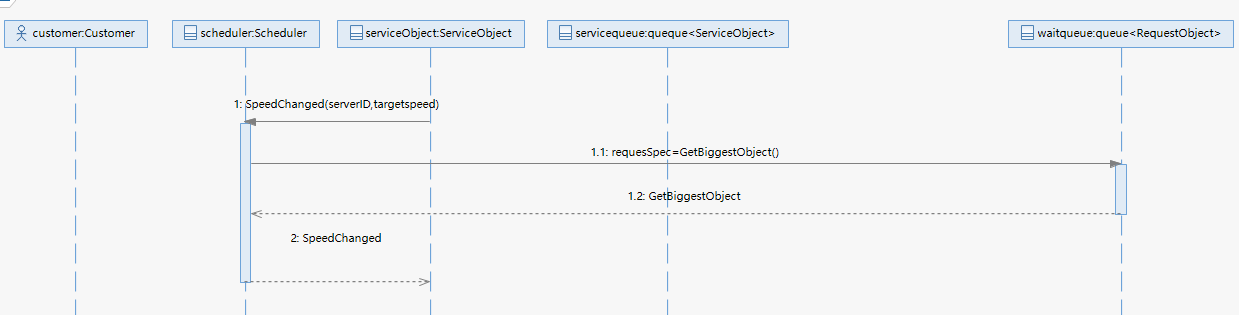
前置条件：客户调用ChangeTargetSpeed方法修改服务队列中服务对象的风速，优先级发生了改变。

调度策略：从等待队列中找到风速比改变的服务对象风速大的最多、时间片最短的请求对象，将服务对象分配给他，原来的服务对象分配一个时间片进入等待队列。没有满足条件的请求对象就没有变化。

时序图：分为3部分



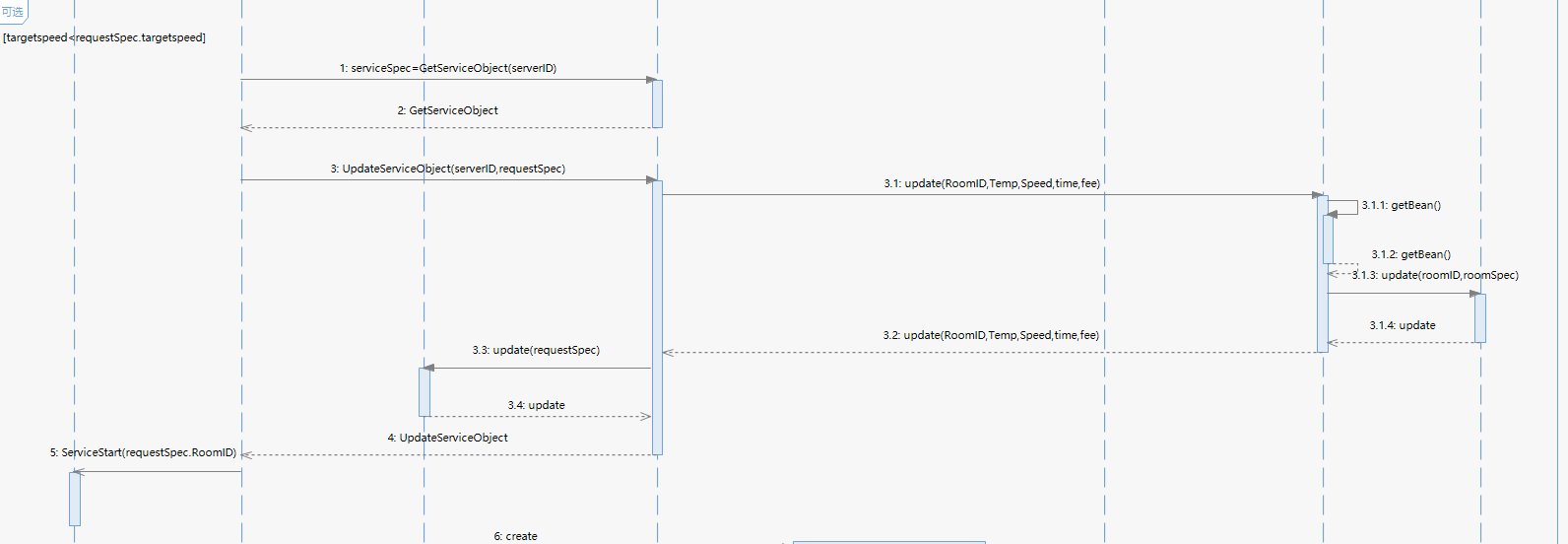
1、服务队列中的风速改变



1. 服务队列向调度对象发出请求消息SpeedChanged,附带改变的服务对象的serverID和targetSpeed。
2. 从等待队列中获取风速最大、时间片最短的对象（优先级最大）getBiggetObject的信息spec。

2、2 3的时序图建立在满足调度条件的情况下，即存在更高优先级的请求对象。

给请求对象分配服务对象。



1. 使用getserviceobject方法通过serverID找到服务对象的信息。
2. 调用updateServiceObject,参数是希望更新的服务对象的ID和请求对象的信息。
3. 服务队列内部调用update方法，对服务对象进行更新，实际上就是为请求对象分配了一个服务对象。
4. 在更新前，首先使用dbfacade的update的方法对持久化层进行更新，将停止服务的服务对象的数据存储数据库中。
5. 然后从请求对象的信息中获取RoomID,向该房间的客户发送信息ServiceStart,说明该房间开始服务。

3、将服务对象放入等待队列中。



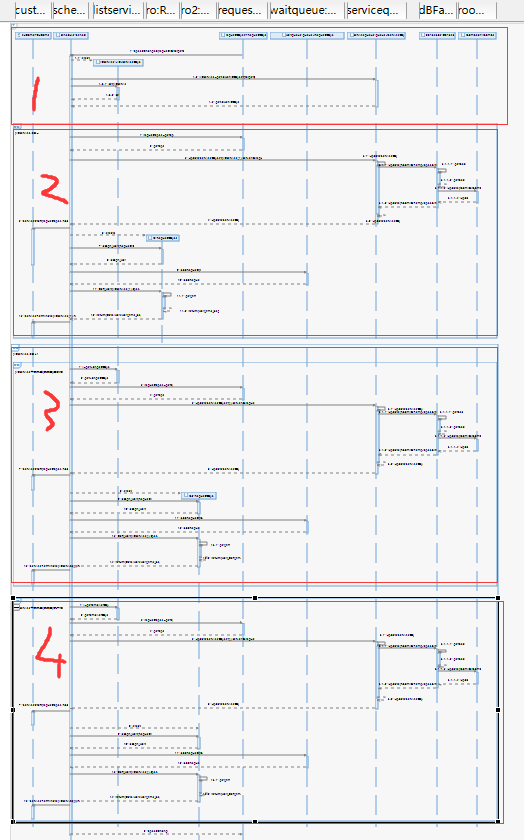
1. 创建请求对象object，并且使用RequestID标志其在队列中的位置。
2. 将object置于等待队列中。
3. 使用房间信息和时间片为object赋值。
4. 根据房间信息的RoomID找到该房间并且给客户回一个ServiceTerminate，说明服务以及终止，进入等待状态，并且附带当前最后的费用信息。

### 4.5.4 客户修改等待队列中的风速

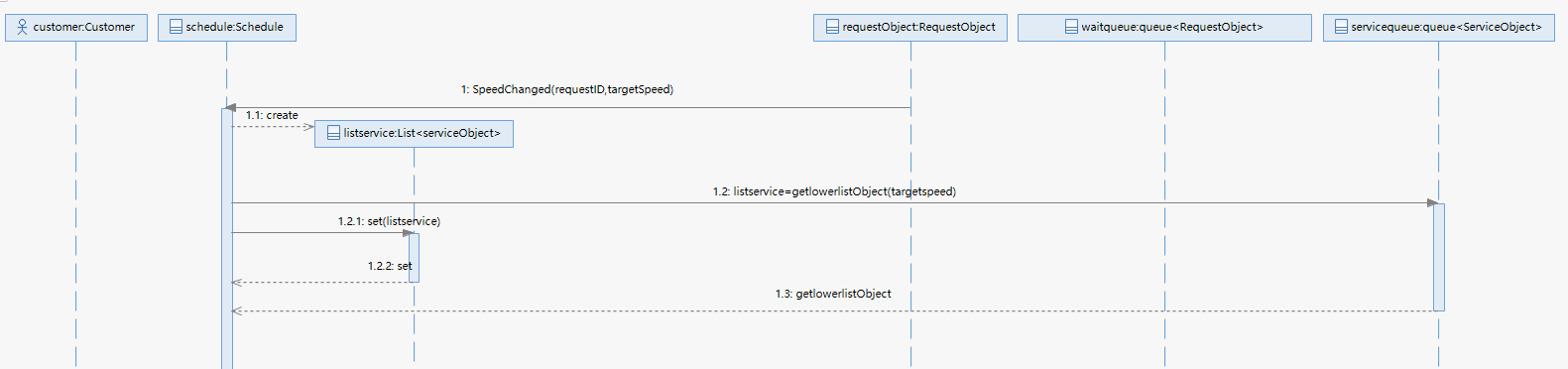
前置条件：客户修改等待队列请求对象中的风速（优先级），需要重新调度该对象。

调度策略：从等待队列中找到所有风速低于该请求对象的服务对象，组成列表。如果该列表为空，说明不存在优先级比该请求对象低的服务对象，则无调度行为。若该列表长度为1，则直接获得列表中对象的服务对象，分配给该请求对象，然后服务对象进入等待队列。如果列表长度大于1，并且列表中的风速都不相等，则将风速最低的服务对象分配给该请求对象；若该队列的风速全都相等，则将该列表中服务时间最长的对象的服务对象分配给该请求对象。

时序图： 由于该调度策略最复杂，需要4张小图来讲解。

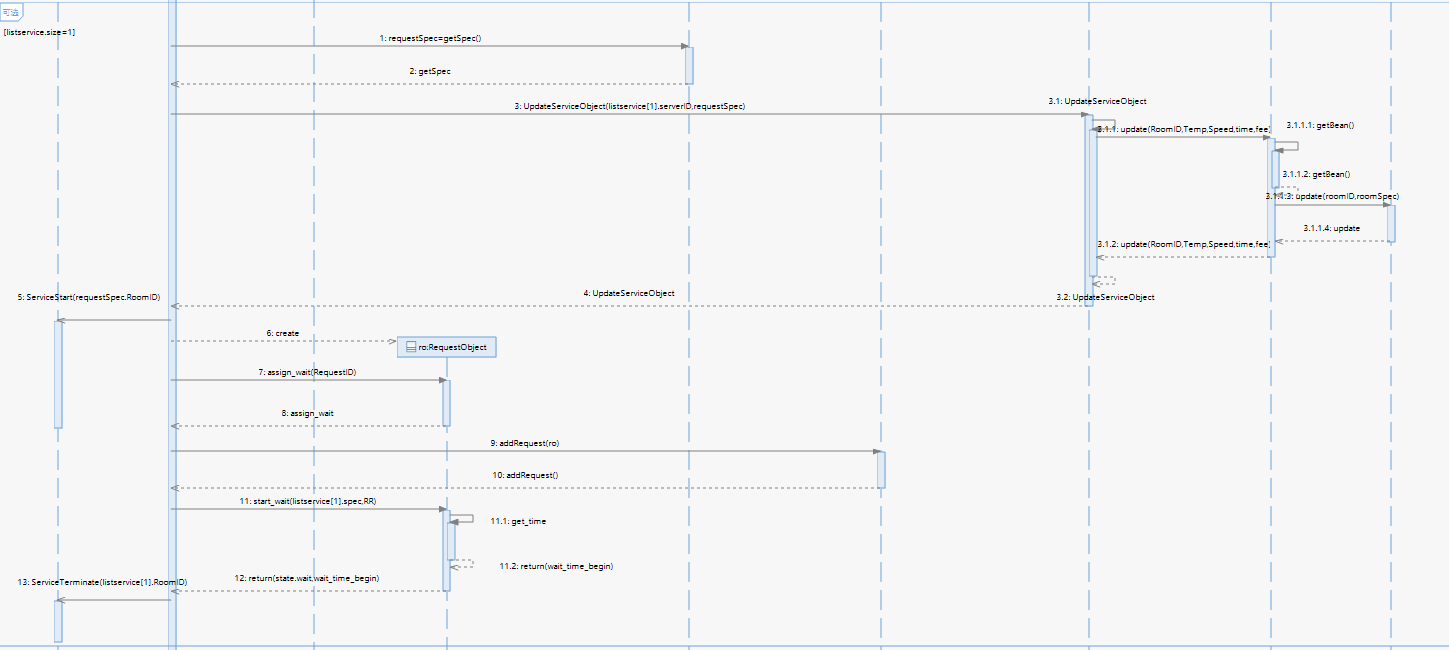


1、客户修改等待队列中的优先级



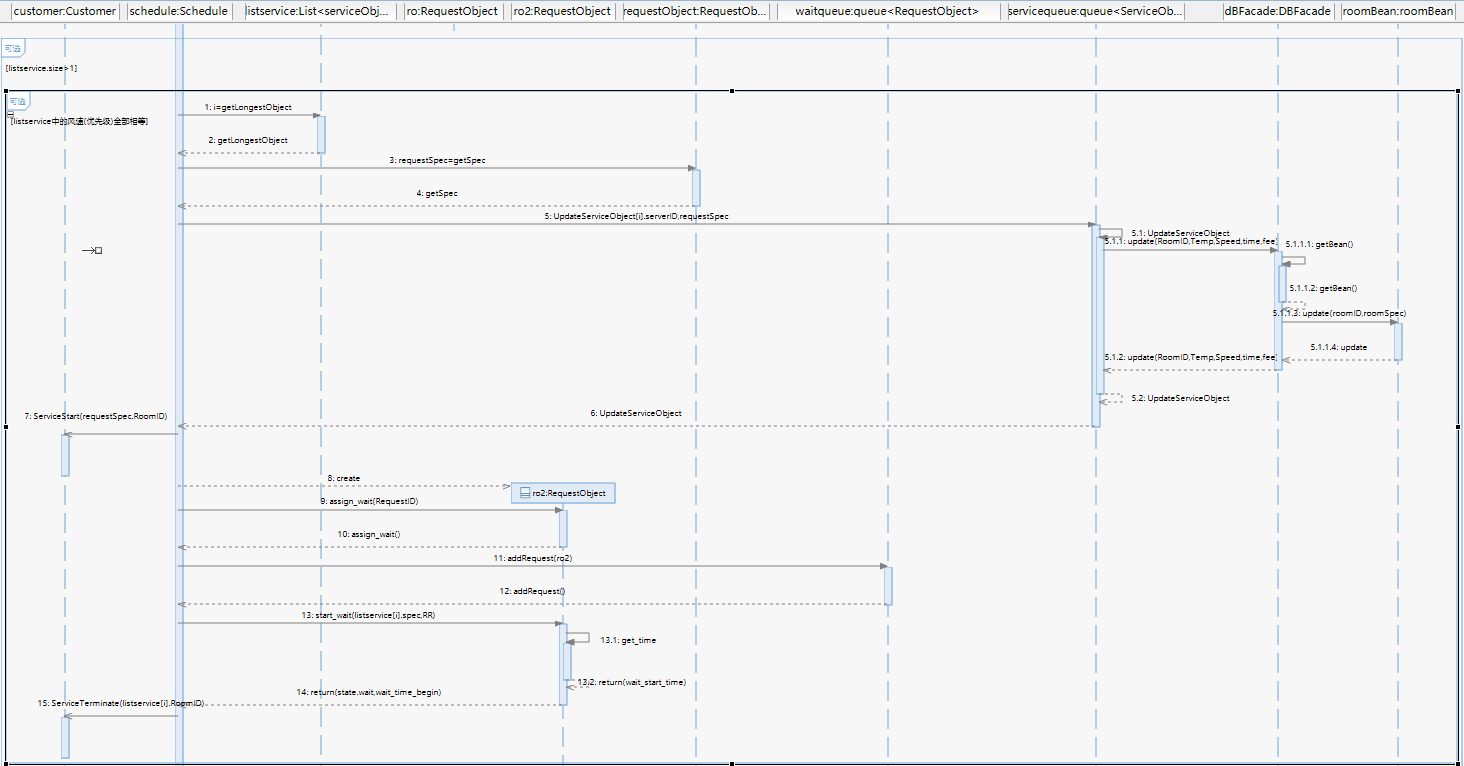
1. 等待队列中优先级变化的请求对象向调度对象发送SpeedChanged，携带RequetsID和targetSpeed（优先级）。
2. 调度对象创建listservice对象。
3. 调度对象使用getLowerListObject方法，返回所有风速比targetspeed低的服务对象。
4. 设置listservice存储这些服务对象的信息。

2、如果列表长度为1



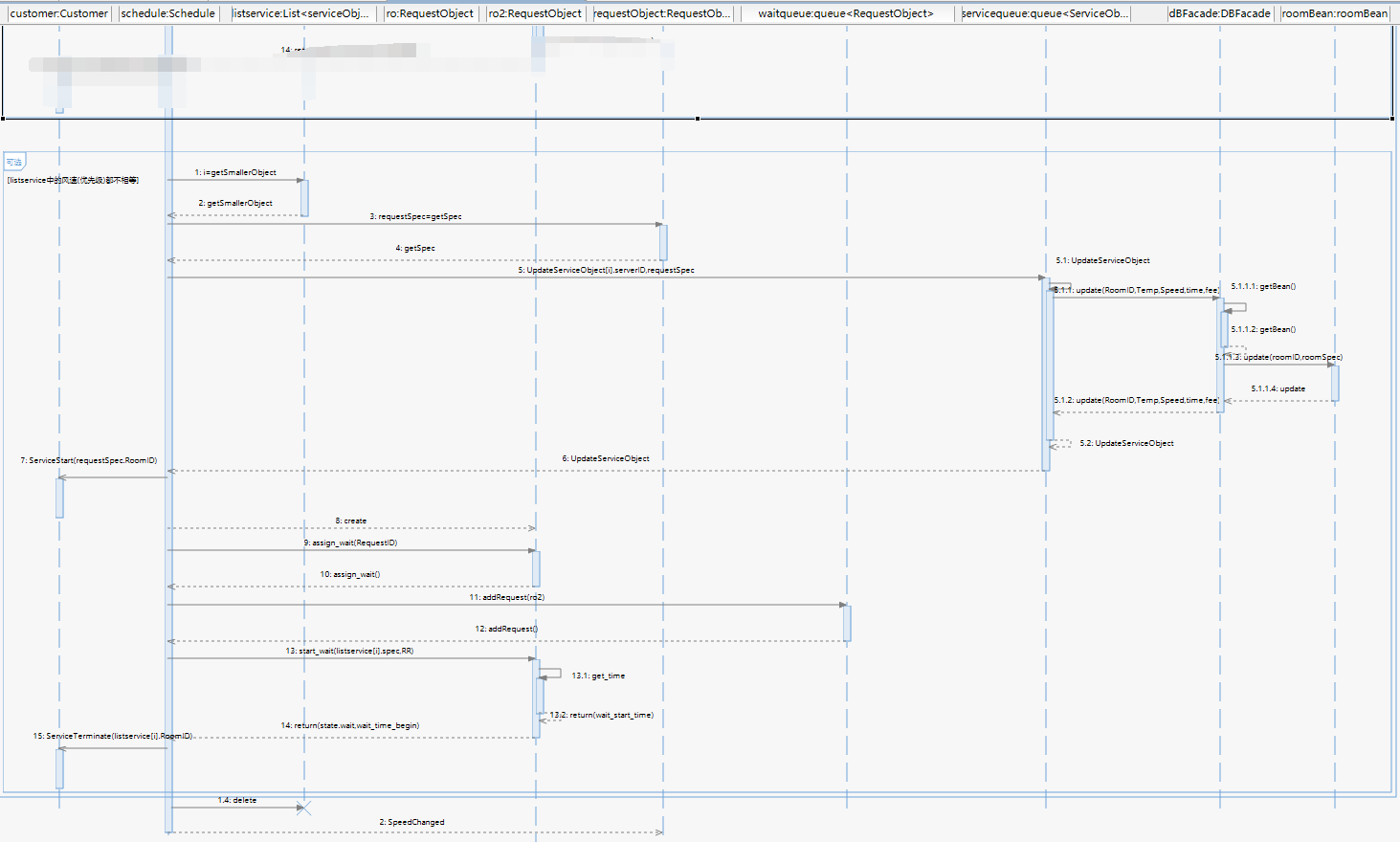
1. 使用getspec方法获取发生改变的请求对象的信息。
2. 调用updateServiceObject,参数是希望更新的服务对象的ID和请求对象的信息。
3. 服务队列内部调用update方法，对服务对象进行更新，实际上就是为请求对象分配了一个服务对象。
4. 在更新前，首先使用dbfacade的update的方法对持久化层进行更新，将停止服务的服务对象的数据存储数据库中。
5. 然后从请求对象的信息中获取RoomID,向该房间的客户发送信息ServiceStart,说明该房间开始服务。
6. 创建请求对象object，并且使用RequestID标志其在队列中的位置。
7. 将object置于等待队列中。
8. 使用房间信息和时间片为object赋值。
9. 根据房间信息的RoomID找到该房间并且给客户回一个ServiceTerminate，说明服务以及终止，进入等待状态，并且附带当前最后的费用信息。

3、 如果列表长度不为1并且列表中的风速全部相等。



1. 调度对象使用getlongestobject方法从listservice获取最大服务时长的服务对象的信息。
2. 调度对象使使用getspec方法获取发生改变的请求对象的信息。
3. 调用updateServiceObject,参数是希望更新的服务对象的ID和请求对象的信息。
4. 服务队列内部调用update方法，对服务对象进行更新，实际上就是为请求对象分配了一个服务对象。
5. 在更新前，首先使用dbfacade的update的方法对持久化层进行更新，将停止服务的服务对象的数据存储数据库中。
6. 然后从请求对象的信息中获取RoomID,向该房间的客户发送信息ServiceStart,说明该房间开始服务。
7. 创建请求对象object，并且使用RequestID标志其在队列中的位置。
8. 将object置于等待队列中。
9. 使用房间信息和时间片为object赋值。
10. 根据房间信息的RoomID找到该房间并且给客户回一个ServiceTerminate，说明服务以及终止，进入等待状态，并且附带当前最后的费用信息

4、 如果列表长度不为1并且列表中的风速全不相等



1. 调度对象使用getsmallerObject方法从listservice获取风速最小的服务对象的信息。
2. 调度对象使使用getspec方法获取发生改变的请求对象的信息。
3. 调用updateServiceObject,参数是希望更新的服务对象的ID和请求对象的信息。
4. 服务队列内部调用update方法，对服务对象进行更新，实际上就是为请求对象分配了一个服务对象。
5. 在更新前，首先使用dbfacade的update的方法对持久化层进行更新，将停止服务的服务对象的数据存储数据库中。
6. 然后从请求对象的信息中获取RoomID,向该房间的客户发送信息ServiceStart,说明该房间开始服务。
7. 创建请求对象object，并且使用RequestID标志其在队列中的位置。
8. 将object置于等待队列中。
9. 使用房间信息和时间片为object赋值。
10. 根据房间信息的RoomID找到该房间并且给客户回一个ServiceTerminate，说明服务以及终止，进入等待状态，并且附带当前最后的费用信息

# 5 分工说明

|  |  |
| --- | --- |
| 成员 | 分工 |
| 程年智 2017211452 | 完成动态结构设计文档的撰写；调度部分设计 |
| 刘凡 2017211442 | 客户开机；客户改变温度、风速；客户关机的结构设计 |
| 高进 2017211520 | 创建详单；打印详单；创建账单；打印账单 |
| 李卓 2017211428 | 打开空调；设置启动参数；启动空调服务，监视空调 |
| 李玉慧 2017211410 | 客户定时收到服务器发来的费用信息；经理查看报表；打印报表，删除报表 |