

动态结构设计v2.0

动态结构设计v2.0

所采用的模型

MVT

经理

对象设计: QueryReport(list_RoomId,type_Report,date)

对象设计: PrintReport(ReportId,date)

前台

对象设计: ReceptionLogin(Password)

对象设计: CreateRDR(RoomNo,BeginDate,EndDate)

对象设计: PrintRDR()

对象设计: CreateBill(RoomNo,BeginDate,EndDate)

对象设计: PrintBill()

管理员

对象设计: PowerOn()

对象设计:

setPara(Mode,Temp_highLimit,Temp_lowLimit,default_TargetTemp,FeeRate_H,FeeRate_M,FeeRate_L)

对象设计: StartUp()

对象设计: CheckRoomState(list_Room)

顾客

对象设计: RequestOn(RoomId,CurrentRoomTemp)

对象设计: ChangeTargetTemp(RoomId,TargetTemp)

对象设计: ChangeFanSpeed(RoomId,FanSpeed)

对象设计: SetMode(RoomId,Mode)

对象设计: RequestOff(RoomId)

调度策略

关于优先级的理解如下:

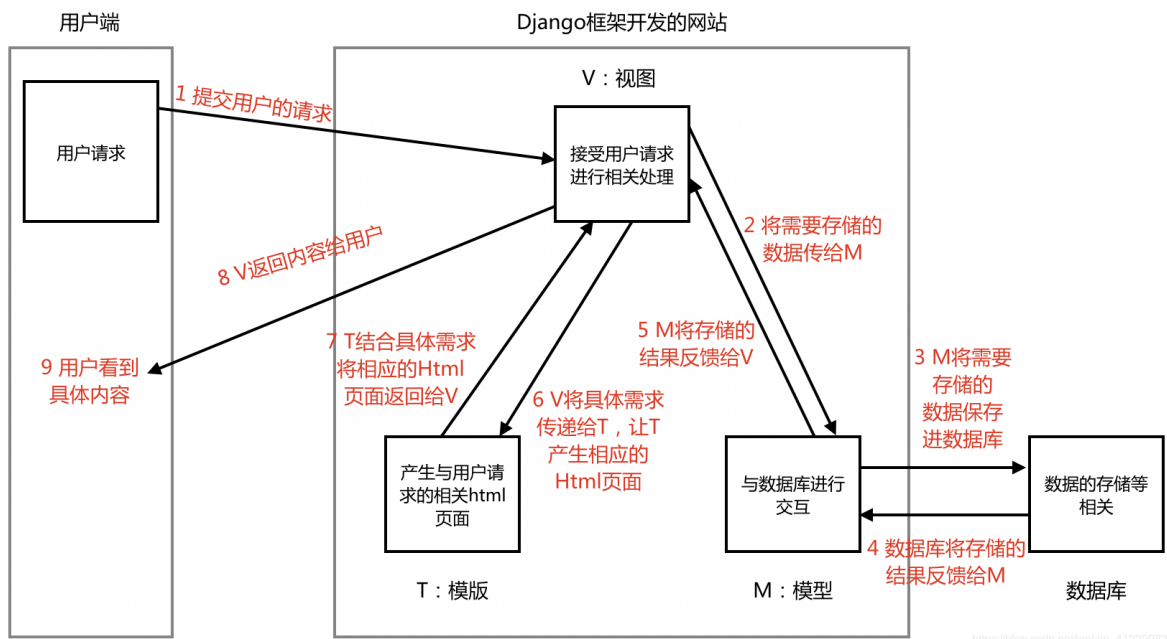
关于时间片调度

人员分配情况

1 所采用的模型

1.1 MVT

(1) Django MVT框架图



(2) 具体功能介绍:

- **M (Model层)** : 与MVC中的M功能相同, 负责和数据库交互, 进行数据处理。
 - (课上说的持久化层也是在这里)
- **V (View层, 业务逻辑层)** : 接收请求, 进行业务处理, 返回应答。
- **url.py (Controller, 控制器层)** , 与MVC中的C功能相同, 负责接受用户请求并分发对应的视图。
 - 当然这个url层是我们自己定义的, 我们认为url.py充当的是一个路由的角色, 书本上对于控制器的标准定义是: 主要用于接受界面层提交的请求并管理和分析这些请求业务的类型, 进而转发给业务逻辑层的业务对象。
- **T (Template, 用户界面GUI层)** : 与MVC中的V功能相同, 负责封装构造要返回的html。

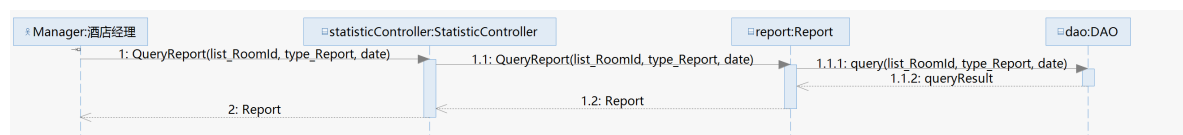
2 经理

在经理这一角色中, StatisticController担任控制器的角色

2.1 对象设计: QueryReport(list_RoomId,type_Report,date)

操作契约:

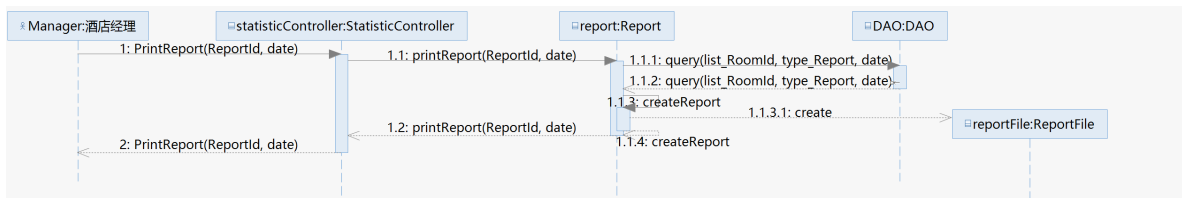
- 1、创建一个统计对象实例;
- 2、创建报表实例;
- 3、修改报表实例的属性;



2.2 对象设计: PrintReport(ReportId,date)

操作契约:

报表文件被创建;

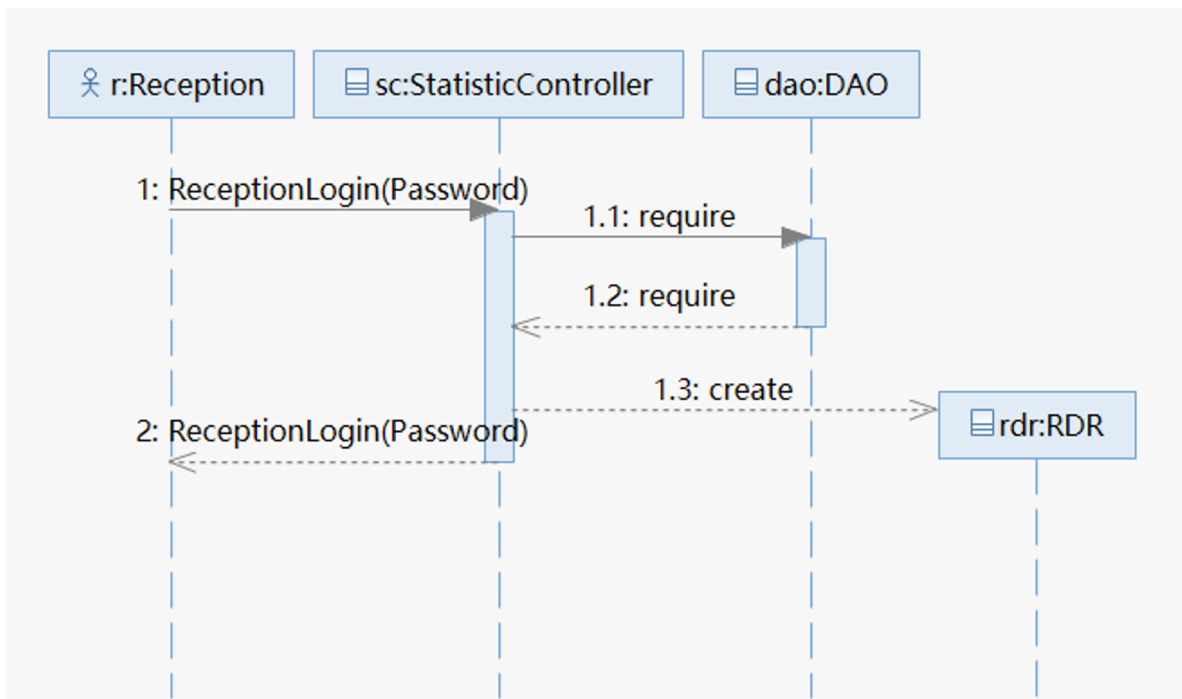


3 前台

在前台的角色当中，StatisticController担任控制器的角色。

3.1 对象设计：ReceptionLogin(Password)

功能：前台登录系统



注：

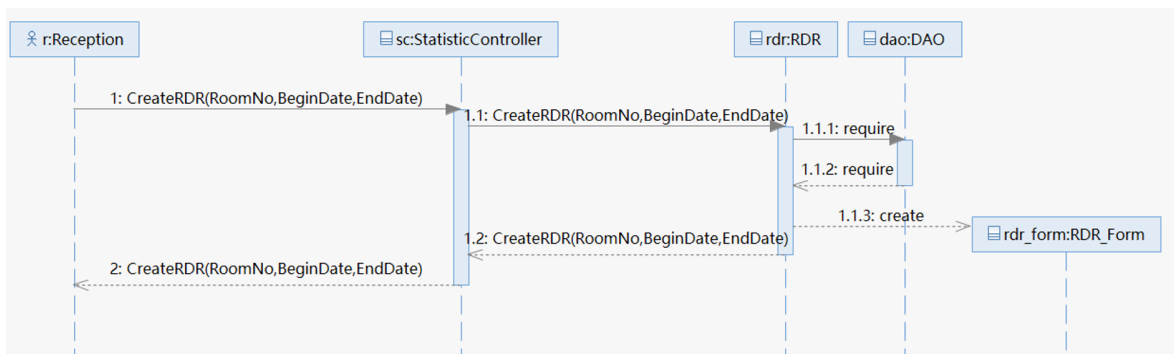
- DAO=Data Access Object，用于与数据库进行数据的存取
- RDR=RequestDetailRecords

3.2 对象设计：CreateRDR(RoomNo,BeginDate,EndDate)

功能：根据房间号、时间来创建详单

操作契约：

- 1、服务对象被创建；
- 2、详单对象被创建；
- 3、详单的所有属性被赋值；



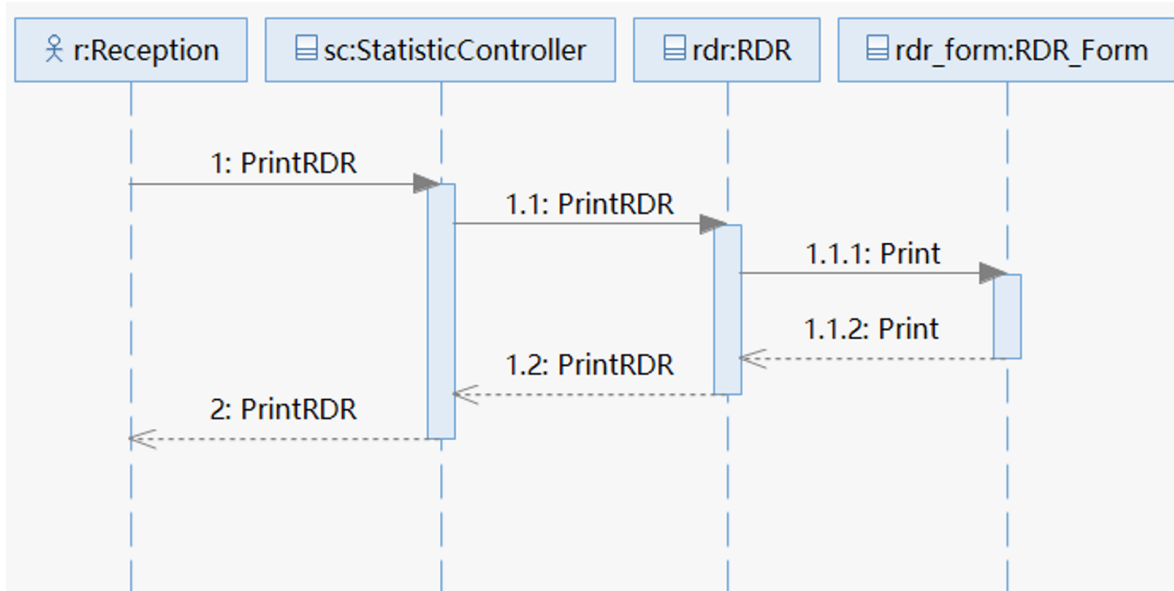
注：RDR_Form是详单文件

3.3 对象设计：PrintRDR()

功能：打印刚创建的详单

操作契约：

1、详单文件被创建；

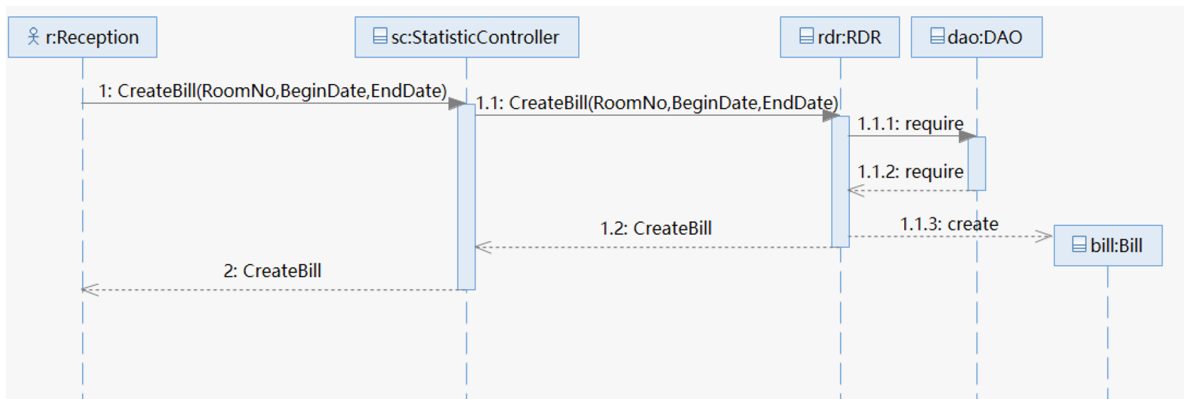


3.4 对象设计：CreateBill(RoomNo,BeginDate,EndDate)

功能：根据房间号、时间来创建账单

操作契约：

- 1、服务对象被创建；
- 2、账单对象被创建；
- 3、账单对象的所有属性被赋值；



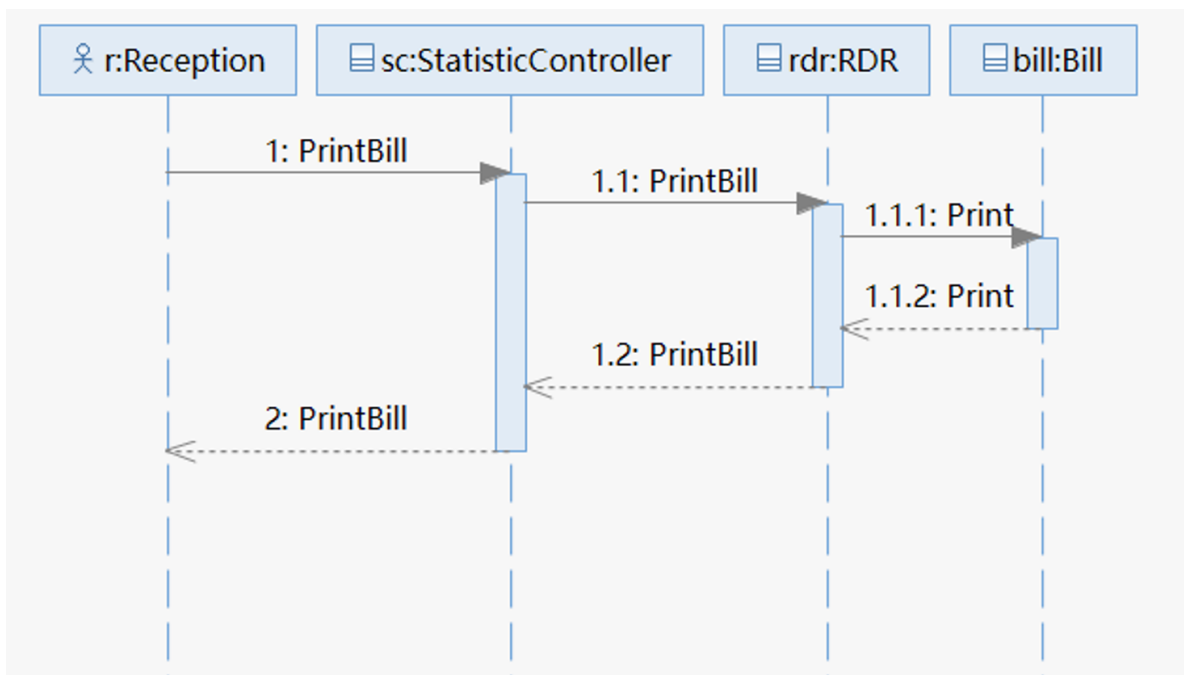
注：Bill是账单文件

3.5 对象设计：PrintBill()

功能：打印刚创建的账单

操作契约：

1、账单文件被创建；



4 管理员

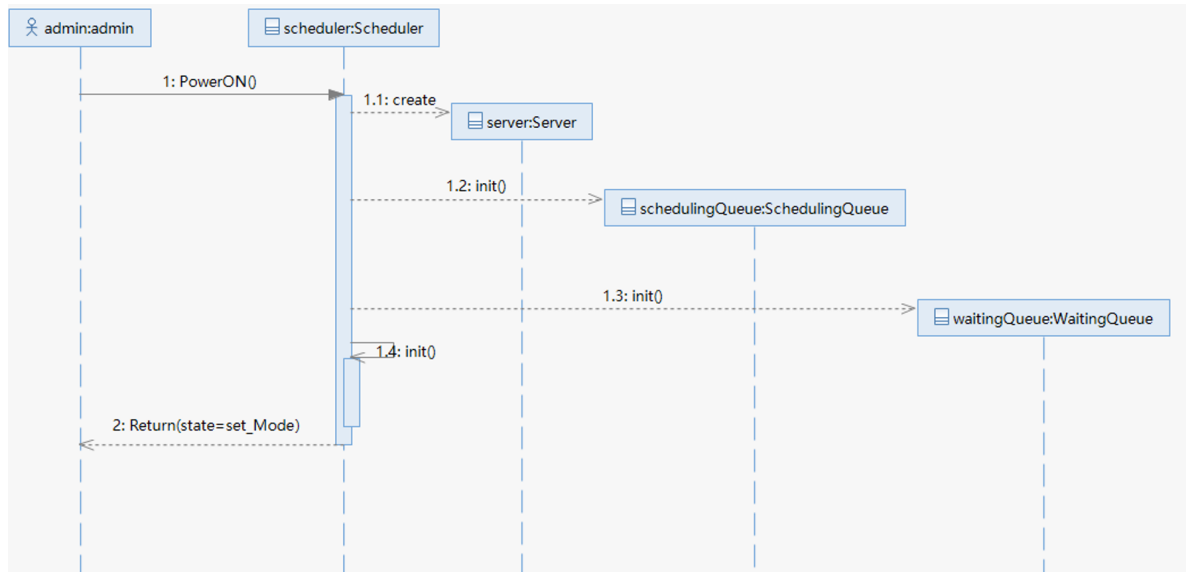
在管理员角色当中，Scheduler调度对象担任控制器的角色，原因如下：

- 调度对象负责接受来自管理员的请求，并转发到对应的业务对象进行操作。

4.1 对象设计：PowerOn()

操作契约：

- 1、服务对象及调度对象被创建；
- 2、调度对象的属性初始化；
- 3、服务队列及等待队列初始化；

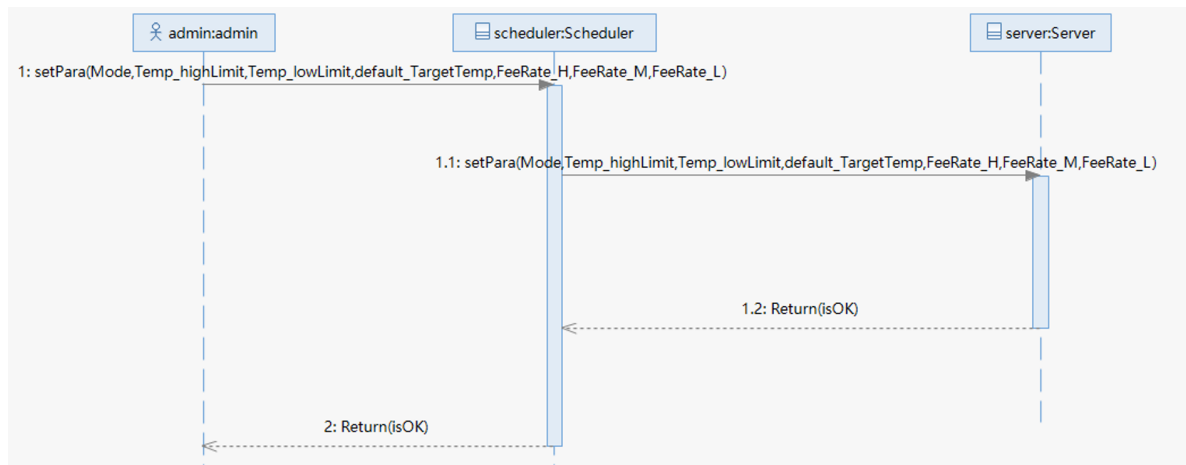


4.2 对象设计:

setPara(Mode,Temp_highLimit,Temp_lowLimit,default_TargetTemp,FeeRate_H,FeeRate_M,FeeRate_L)

操作契约:

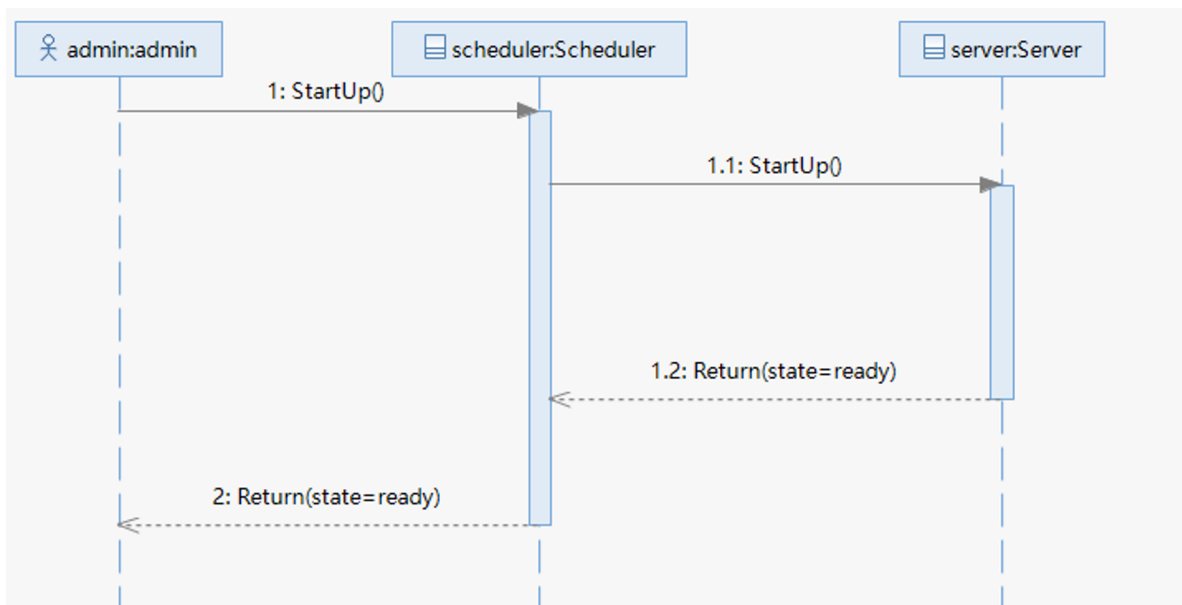
- 1、服务对象的属性初始化



4.3 对象设计: StartUp()

操作契约:

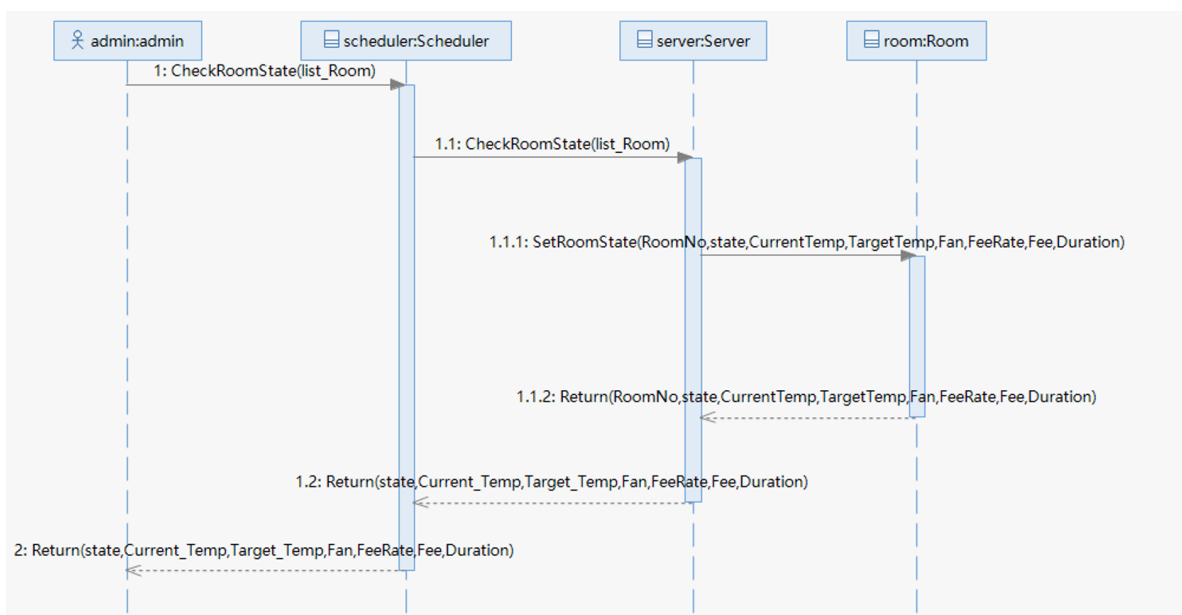
- 1、服务对象及调度对象的状态被设置为: 就绪



4.4 对象设计：CheckRoomState(list_Room)

操作契约：

- 1、与调度对象建立关联；
- 2、与服务对象建立关联；
- 3、修改房间的状态信息；



5 顾客

5.1 对象设计：RequestOn(RoomId,CurrentRoomTemp)

操作契约：

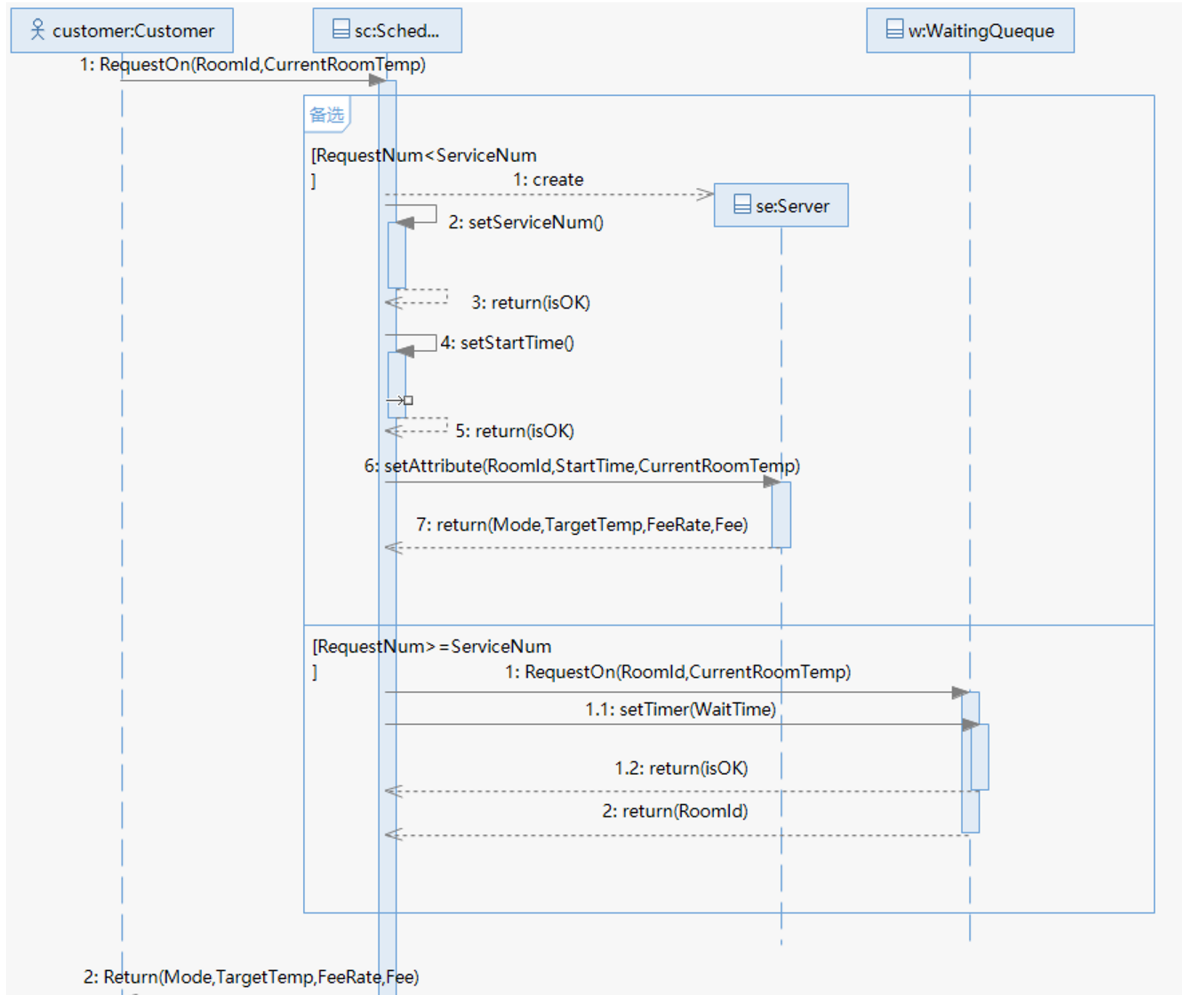
If (RequestNum<ServiceNum)

- 1、调度对象与房间建立关联；
- 2、一个服务对象被创建（当前服务对象数小于服务对象数上限，验收环境的服务对象上限数=3）；
- 3、调度对象与服务对象建立关联；

- 4、服务对象与房间建立关联；
- 5、调度对象的服务对象数及服务开始时间被赋值；
- 6、服务对象的服务状态，服务开始时间，模式，目标温度，费率及费用值被赋值；

else

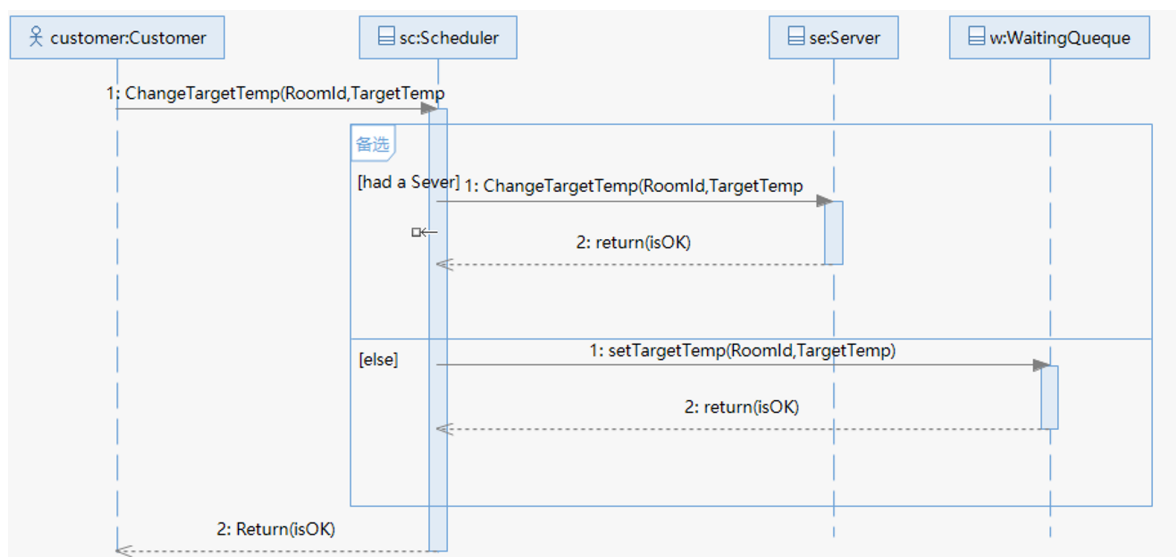
- 1、调度对象与房间建立关联；
- 2、当前服务对象数大于等于服务对象数上限，则将房间的请求放到等待队列进行调度；
- 3、队列中的等待服务的房间的等待时长被赋值（时间片时长）；
- 4、调度对象保存



5.2 对象设计：ChangeTargetTemp(RoomId,TargetTemp)

操作契约：

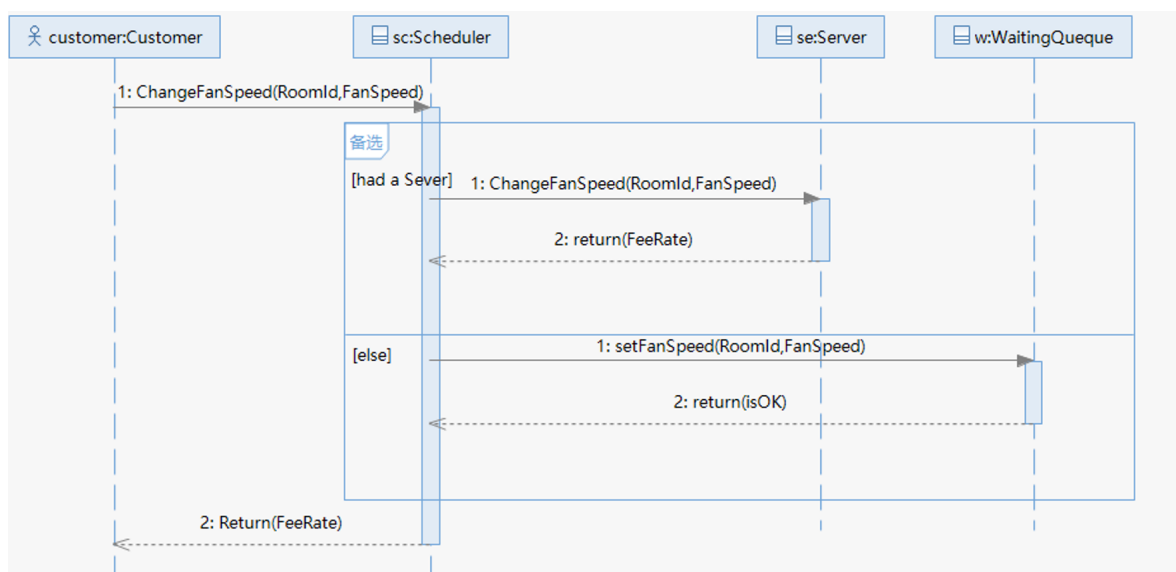
- 1、调度对象与房间建立关联（调度对象首先接受调温的请求）；
- 2、如果该请求的房间有服务对象，则调度对象与服务对象建立关联（调度对象将请求转发给服务对象）；
- 3、如果该请求的房间在等待队列，则调度对象修改等待队列中的目标温度属性TargetTemp；



5.3 对象设计： ChangeFanSpeed(RoomId,FanSpeed)

操作契约：

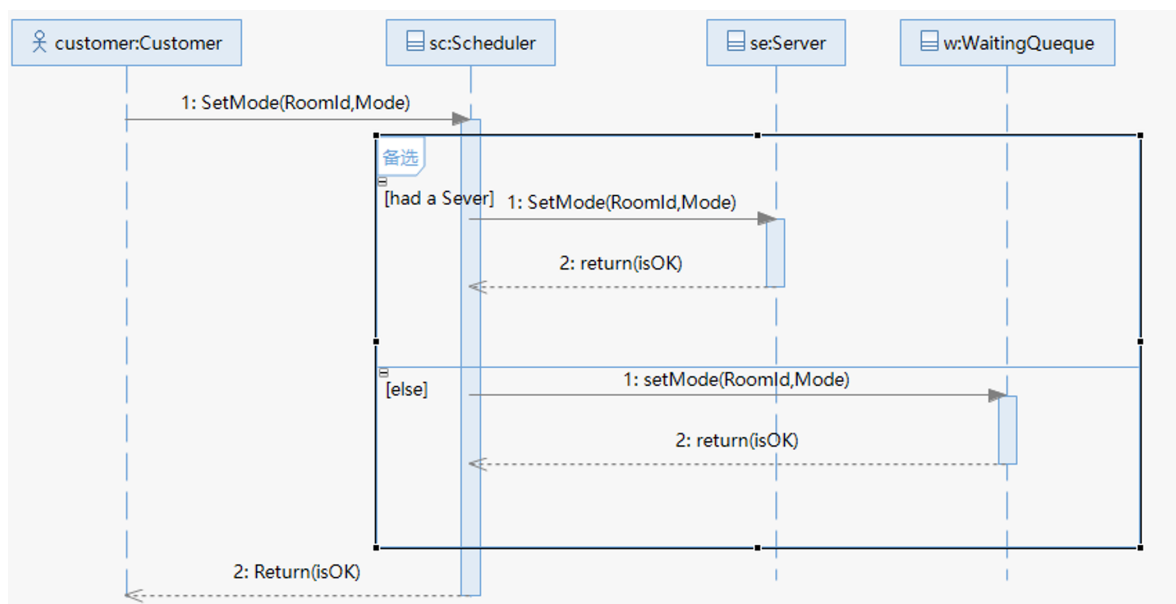
- 1、调度对象与房间建立关联（调度对象首先接受调风请求）；
- 2、如果该请求的房间有服务对象，则调度对象与服务对象建立关联（调度对象将请求转发给服务对象）；
- 3、如果该请求的房间在等待队列，则调度对象修改等待队列中的目标风速属性FanSpeed；



5.4 对象设计： SetMode(RoomId,Mode)

操作契约

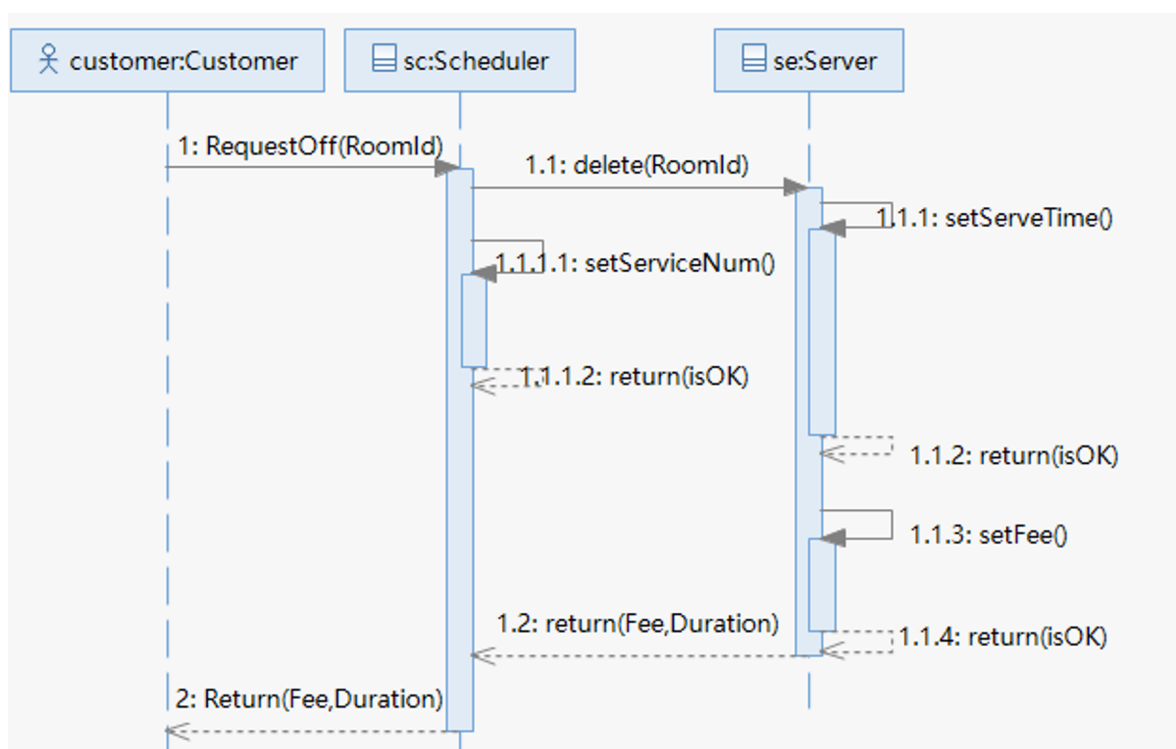
- 1、调度对象与房间建立关联（调度对象首先接受调模式的请求）；
- 2、如果该请求的房间有服务对象，则调度对象与服务对象建立关联（调度对象将请求转发给服务对象）；
- 3、如果该请求的房间在等待队列，则调度对象修改等待队列中的目标模式属性Mode；



5.5 对象设计：RequestOff(RoomId)

操作契约：

- 1、调度对象与房间删除关联；
- 2、调度对象的服务对象数被赋值；
- 3、服务对象与房间删除管理；
- 4、服务对象的服务时长和费用被修改；



6 调度策略

如果当前请求空调服务的人数大于3时，将采用一些调度策略

- 优先级
- 时间片

6.1 关于优先级的理解如下：

如果当前服务队列已经满载，而此时突然来了一个请求，该请求存在如下条件之一

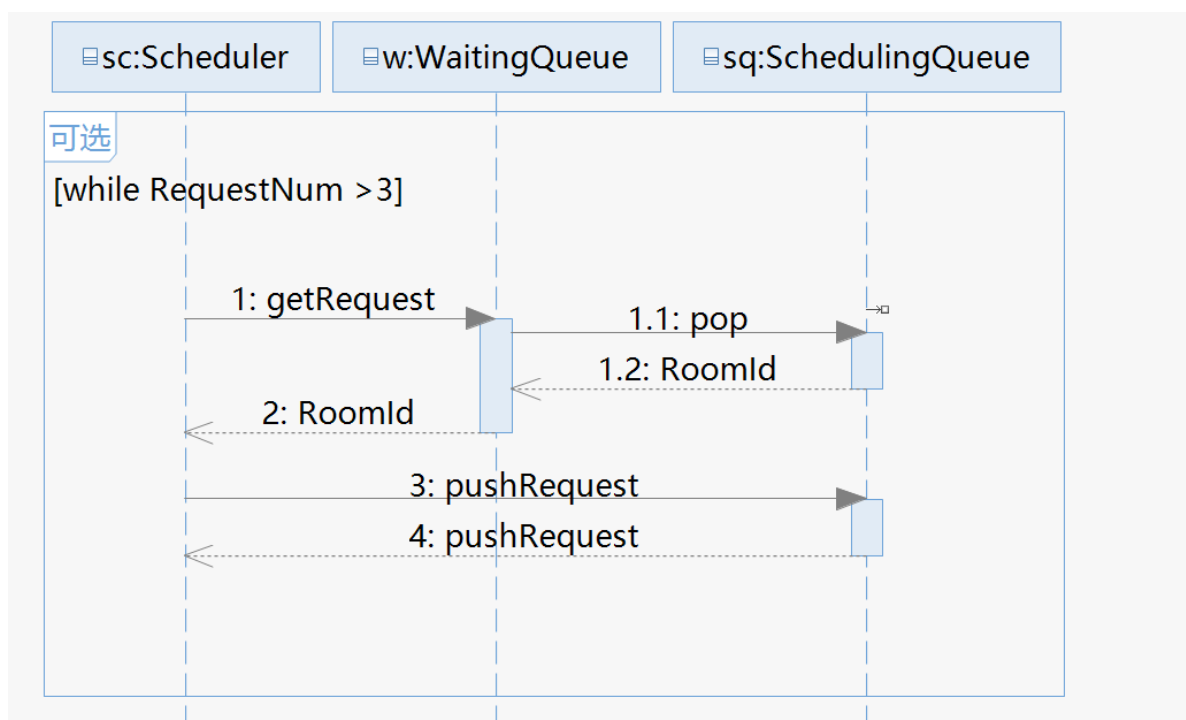
- 风速为高风速，
- 温度小于26
- 模式为制冷

那么将以高优先级的形式插队，从服务队列当中抽取优先级最低的服务对象，将其移出服务队列，移入等待队列，所谓的低优先级有如下定义：

- 当前服务时间在服务队列中最长。

6.2 关于时间片调度

如果当前服务队列已经满载，而此时来了一个请求，该请求是普通请求，那么该请求将进入等待队列。并且调度器设置其等待时间为2min。在两分钟后，从服务队列中取出优先级最低的服务对象，从等待队列中取出该请求，将其加入服务队列为其分配服务对象。



7 人员分配情况

组长	陈宇琛	负责经理角色，总结并修改组员的时序图，并完成调度部分。文档的编写
组员	张若冰	负责前台角色
组员	车培帅	负责管理员角色
组员	李佳豪	负责顾客角色
组员	孙一博	文档的编写