



**分布式温控系统设计模型**

**动态结构设计**

编写者：张绍磊、王飞扬、李智鑫、常亮、韩易辰、李光

2019年5月12日

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本修订记录 | | | | | |
| **编号** | **日期** | **版本号** | **章节** | **编写者** | **说明** |
| 1 | 2019-5-12 | V1.0 | 文档格式、封面、目录、1、2、3、8 | 张绍磊 | 设计有关调度的交互过程、文档说明、项目背景、汇总 |
| 4 | 王飞扬、韩易辰 | 空调计费系统的动态结构设计——住客 |
| 5 | 李智鑫 | 空调计费系统的动态结构设计——前台 |
| 6 | 常亮、李光 | 空调计费系统的动态结构设计——管理员 |
| 7 | 韩易辰 | 空调计费系统的动态结构设计——经理 |

目录

[1. 文档说明 4](#_Toc8562301)

[1.1 文档目的 4](#_Toc8562302)

[1.2 文档范围 4](#_Toc8562303)

[1.3 读者对象 4](#_Toc8562304)

[1.4 参考文档 4](#_Toc8562305)

[1.5术语与缩写解释 5](#_Toc8562306)

[2. 项目背景 5](#_Toc8562307)

[3. 软件体系结构设计 5](#_Toc8562308)

[4. 动态结构设计——住客 6](#_Toc8562309)

[用例图 6](#_Toc8562310)

[用例实现过程 7](#_Toc8562311)

[1.RequestOn(RoomId,CurrentRoomTemp) 7](#_Toc8562312)

[2.ChangeTargetTemp(RoomId,TargetTemp) 8](#_Toc8562313)

[3.ChangeFanSpeed(RoomId,FanSpeed) 8](#_Toc8562314)

[4.RequestOff(Roomid) 9](#_Toc8562315)

[5.RequestFee 10](#_Toc8562316)

[5. 动态结构设计——前台 11](#_Toc8562317)

[用例图 11](#_Toc8562318)

[用例实现过程 11](#_Toc8562319)

[1.CreateRDR(RoomId,date\_in,date\_out) 11](#_Toc8562320)

[2.PrintRDR(RoomId,list\_RDR) 12](#_Toc8562321)

[3.CreateInvoice(RoomId,date\_in,date\_out) 14](#_Toc8562322)

[4.PrintInvoice(RoomId,date\_out,Toal\_fee) 15](#_Toc8562323)

[6. 动态结构设计——管理员 16](#_Toc8562324)

[用例图 16](#_Toc8562325)

[用例实现过程 16](#_Toc8562326)

[1.PowerON() 16](#_Toc8562327)

[2.setPara(Mode,Temp\_highLimit,Temp\_lowLimit,default\_TargetTemp,FeeRate\_H,FeeRate\_M,FeeRate\_L） 17](#_Toc8562328)

[3.StartUp() 19](#_Toc8562329)

[4.CheckRoomState(list\_Room) 20](#_Toc8562330)

[7. 动态结构设计——经理 21](#_Toc8562331)

[用例图 21](#_Toc8562332)

[用例实现过程 21](#_Toc8562333)

[1.QueryReport(list\_RoomId,type\_Report,date) 21](#_Toc8562334)

[2. PrintReport(ReportId,date) 22](#_Toc8562335)

[8. 动态结构设计——调度交互过程 23](#_Toc8562336)

[调度策略 23](#_Toc8562337)

[调度过程实现 24](#_Toc8562338)

[1.RequestNumber>ServiceNumber 24](#_Toc8562339)

# 1. 文档说明

## 1.1 文档目的

文档主要的目的是实现分布式温控系统的动态结构设计。通过找出与系统交互的外部事物，说明它们如何与系统交互，更易于对系统行为进行探讨和理解。旨在进一步描述客户的需求、建立软件设计的基础，以及定义在软件完成后可以被确认的一组需求，便于用户和软件开发人员共同理解系统框架，为后续产品形成详尽的规格说明。

本软件需求规格说明书的受众面广泛，包括项目产品经理，项目设计人员，项目编码、测试和维护人员，客户等。

## 1.2 文档范围

根据分布式温控系统的用例图以及文本描述进行系统设计，其中包含了4个主要参与者：顾客、前台、空调管理员、经理。首先通过用例图构建出所有可能用例与各个参与者之间的框架结构关系，在搭建起来的UML图的基础上，再逐一对各个用例进行具体分析，分解出每一个用例的细节动作步骤。

## 1.3 读者对象

用户、设计人员、编码人员和测试人员本软件需求规格说明书的受众面广泛，包括项目产品经理，项目设计人员，项目编码、测试和维护人员和详细设计负责人员等。

## 1.4 参考文档

* 《分布式温控系统的用例模型》——310d项目组
* 《小型两级分布式温度控制系统的研究与设计（1997）》——贺玲芳
* 《一种小型分布式工业温度控制器及其算法的研究（1994）》——张彦斌、常弘
* 《一种小型分布式微机控制系统通信控制器的研制（1990）》——宋书中、梁伸
* 《软件工程模型与方法（第2版）》——肖丁、修佳鹏著 北京邮电大学出版社

## 1.5术语与缩写解释

ACSE Association Control Service Element 连接控制服务元素

ALC Automatic Level Control 自动平衡控制

BAC Building Automation & Control net 建筑物自动化与控制网络

BUE betriebsfertige Uebergabe 验收

CAD Computer Aided Design 计算机辅助设计

DER Distributed Energy Resource 分布式供能

FMEA Fehler Moeglichkeits and Einfluss Analyes 缺陷可能性与影响分析

GUI Graphic User Interface 图形用户界面

IDE Integrated Developing Environment 集成开发环境

# 2. 项目背景

某快捷连锁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要求设定温度和风速的调节，同时可以显示所消耗的能量以及所需支付的金额。除此之外，酒店针对每个房间需要给出空调使用的详单，方便客户退房时进行结账，以及酒店空调使用的各式统计报表。

# 3. 软件体系结构设计

系统采用分层的体系结构，交互图设计部分考虑了持久化层的对象设计。其软件分层结构如下所示：



# 4. 动态结构设计——住客

## 用例图



## 用例实现过程

### 1.RequestOn(RoomId,CurrentRoomTemp)

顾客使用空调操作契约：

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；

设计用例：



实现过程：

1. Register作为控制器对象接收系统事件RequestOn
2. Register将服务对象数及服务开始时间赋值给调度对象Dispatcher的实例d
3. 关联Dispatcher和Room
4. 通过Dispatcher创建一个服务对象Service实例s
5. 关联Service和Room
6. Service将服务开始时间、模式、目标温度、费率及费用值赋值给Service的实例s

### 2.ChangeTargetTemp(RoomId,TargetTemp)

调节温度操作契约：

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;

设计用例：



实现过程：

1. Register作为控制器对象接收系统事件ChangeTargetTemp
2. Register将系统事件ChangeTargetTemp传给调度对象Dispatcher
3. 关联Dispatcher和Room
4. 如果该请求的房间有服务对象的实例s，则Dispatcher与Service建立关联
5. 如果该请求的房间在等待队列w中，则调度对象修改等待队列中的TargetTemp属性

### 3.ChangeFanSpeed(RoomId,FanSpeed)

调节风速操作契约：

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风速属性FanSpeed;

设计用例：



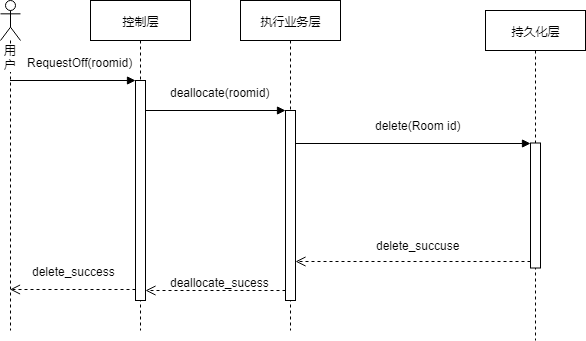
实现过程：

1. Register作为控制器对象接收系统事件ChangeTargetTemp
2. Register将系统事件ChangeTargetTemp传给调度对象Dispatcher
3. 关联Dispatcher和Room
4. 如果该请求的房间有服务对象的实例s，则Dispatcher与Service建立关联
5. 如果该请求的房间在等待队列w中，则调度对象修改等待队列中的TargetTemp属性

### 4.RequestOff(Roomid)

操作契约：

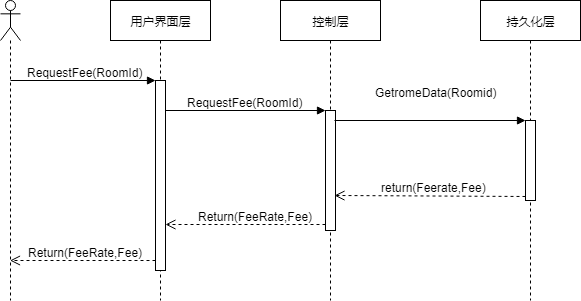
1. 调度对象与房间建立联系
2. 根据roomid查找要删除的对象并进行删除
3. 取消调度对象与房间的联系
4. 取消房间与服务对象的联系



实现过程：

1. 控制层接受用户的删除指令
2. 将要删除的房间信息喜欢给业务层进行执行
3. 删除持久化层中的数据
4. 取消业务层与持久化层的关联
5. 取消控制层与业务层的关联
6. 向用户返回操作结果

### 5.**RequestFee**

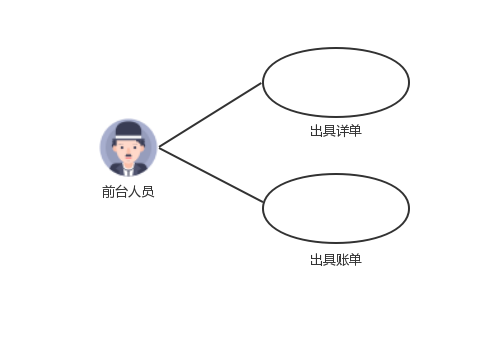


实现过程：

1. 系统每分钟自动发送；
2. 实现与层次说明；
3. 由控制层发出请求，从持久化数据并返回。

# 5. 动态结构设计——前台

## 用例图



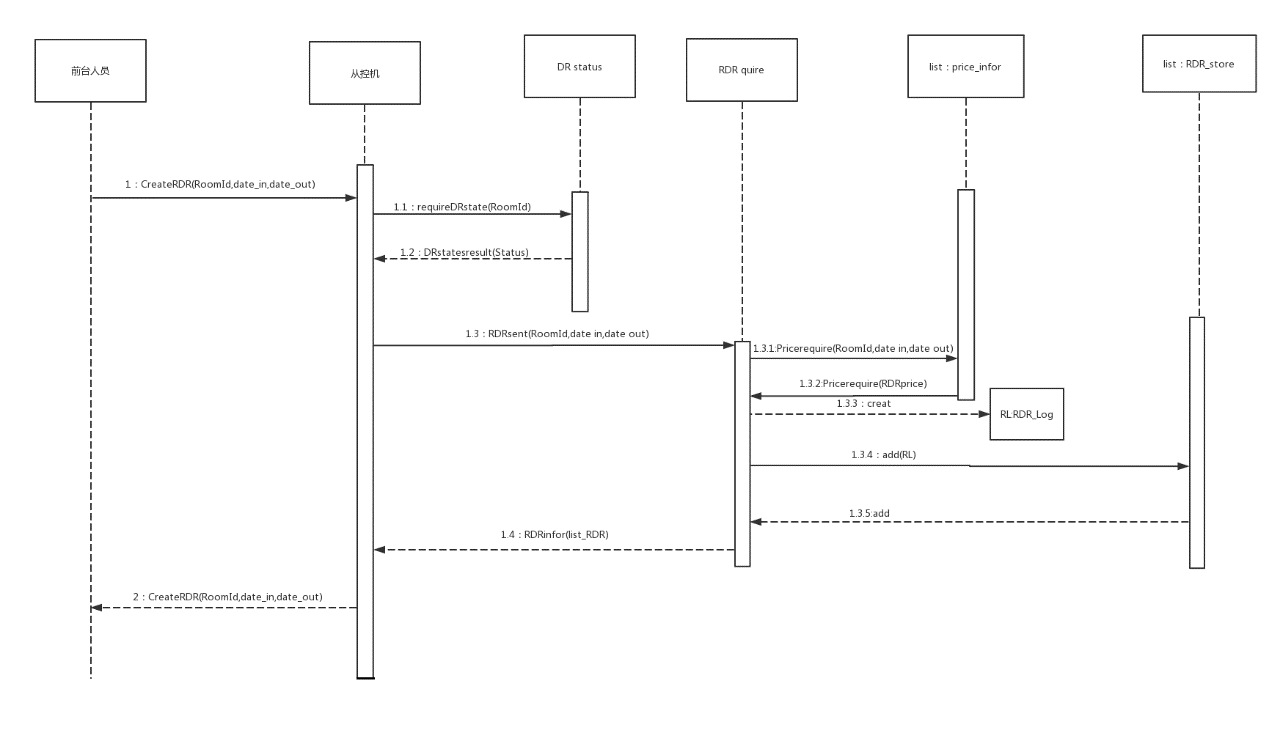
## 用例实现过程

### 1.CreateRDR(RoomId,date\_in,date\_out)

（1）、操作契约

|  |  |
| --- | --- |
| 系统事件 | CreateRDR(RoomId,date\_in,date\_out) |
| 交叉引用 | 酒店前台-从控机 |
| 前置条件 | 详单对象未被创建 |
| 后置条件 | 1. 详单对象被创建 2. 初始化详单对象 |

（2）、交互图



（3）、交互图说明及层次化设计

用户界面层：与前台人员进行交互，用于发出创建详单请求。

控制层：从控机控制层接收请求，检查详单信息是否合法

业务逻辑层：RDR quire，用于实现创建详单的具体功能需求，包括查询计算价格，记录查询操作的日志。

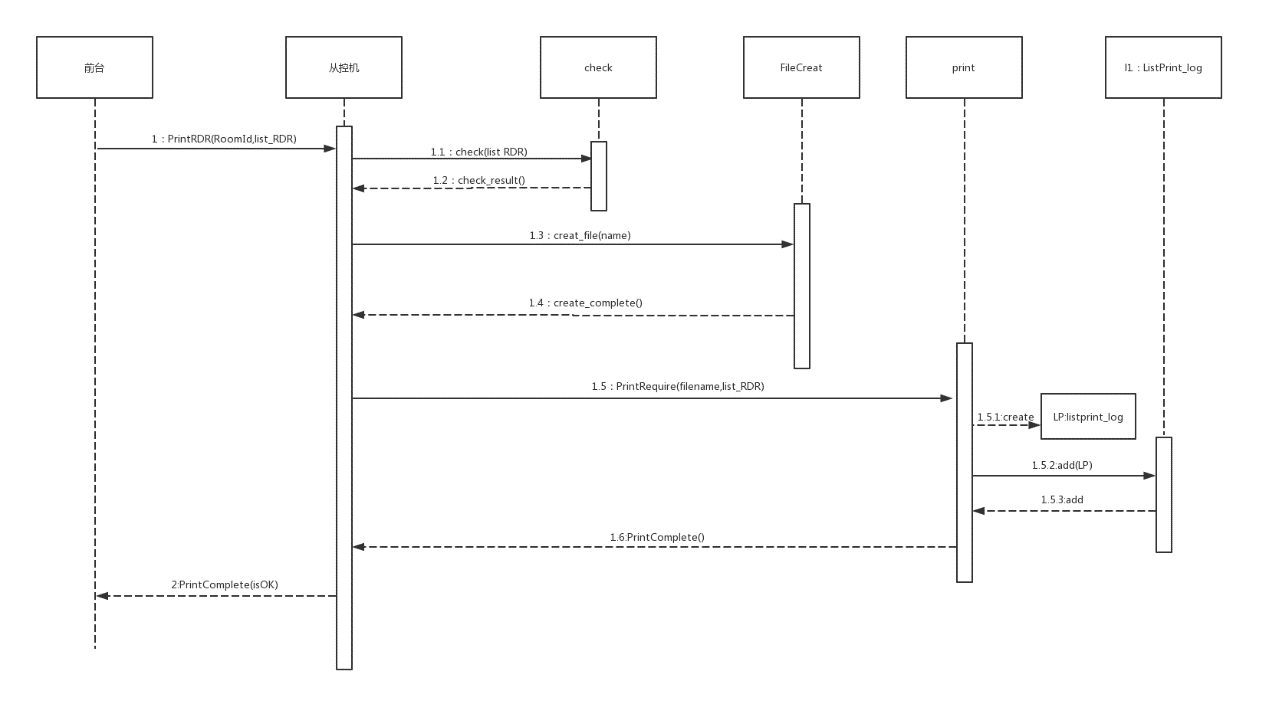
持久化层：RDR\_store，在系统完成创建操作后，将日志信息永久保存于数据库中

### 2.PrintRDR(RoomId,list\_RDR)

（1）、操作契约

|  |  |
| --- | --- |
| 系统事件 | PrintRDR(RoomId,list\_RDR) |
| 交叉引用 | 酒店前台-从控机 |
| 前置条件 | 系统收到详单信息 |
| 后置条件 | 输出详单文件 |

（2）、交互图



（3）、交互图说明及层次化设计

用户界面层：与前台人员进行交互，用于发出打印详单请求。

控制层：从控机作为控制层接收请求，检查详单信息是否合法，而后发出创建文件以及打印请求

业务逻辑层：print，用于实现创建打印详单的具体功能需求，包括打印详单和记录打印日志

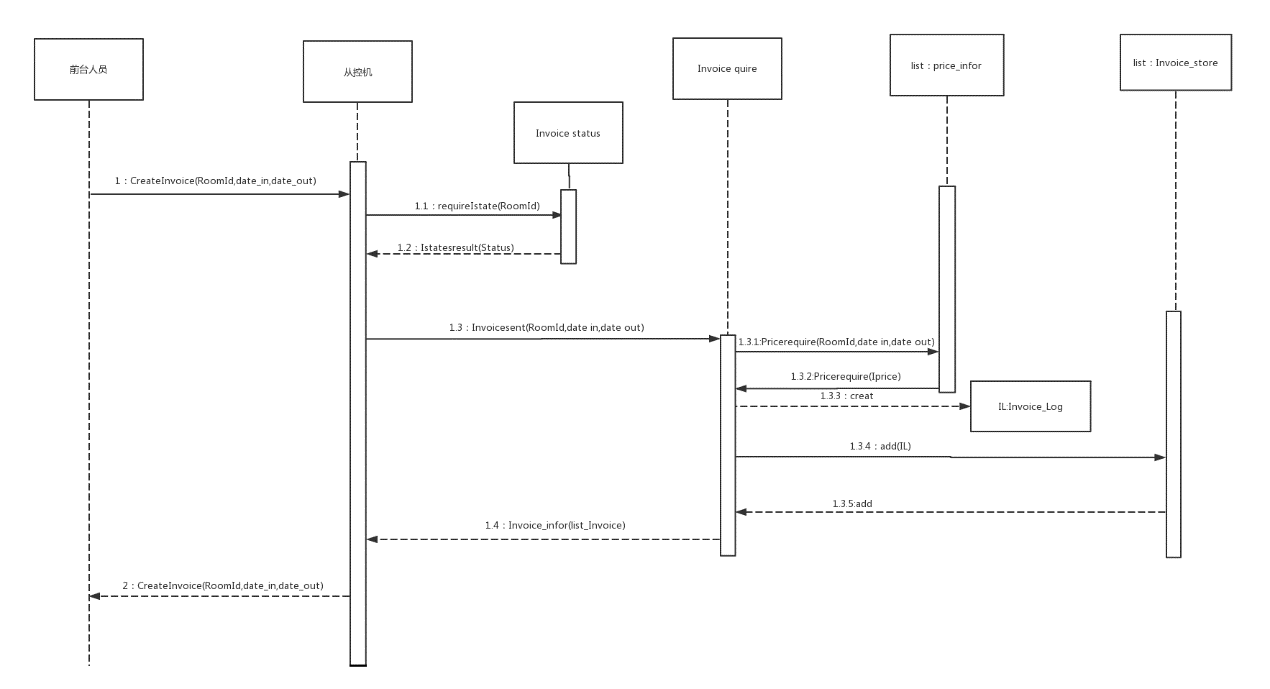
持久化层：ListPrint\_Log，在打印操作完成后，将打印操作的日志信息永久保存于数据库中

### 3.CreateInvoice(RoomId,date\_in,date\_out)

（1）、操作契约

|  |  |
| --- | --- |
| 系统事件 | CreateInvoice(RoomId,date\_in,date\_out) |
| 交叉引用 | 酒店前台-从控机 |
| 前置条件 | 账单对象未被创建 |
| 后置条件 | 1. 创建账单对象 2. 初始化账单对象值 |

（2）、交互图



（3）、交互图说明及层次化设计

用户界面层：与前台人员进行交互，用于发出出账单请求。

控制层：从控机控制层接收请求，检查账单信息是否合法

业务逻辑层：Invoice quire，用于实现出账单的具体功能需求，包括查询计算价格，记录出账单的操作日志。

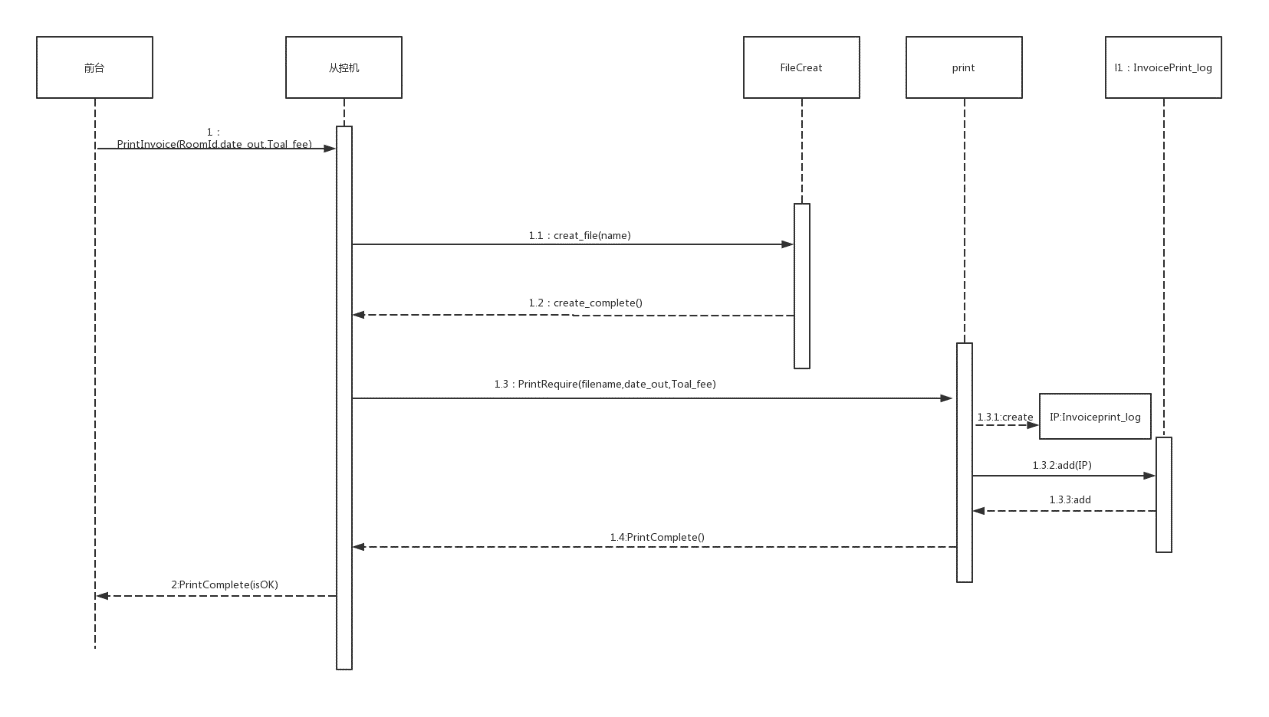
持久化层：Invoice\_store，在系统完成创建操作后，将日志信息永久保存于数据库中

### 4.PrintInvoice(RoomId,date\_out,Toal\_fee)

（1）、操作契约

|  |  |
| --- | --- |
| 系统事件 | PrintInvoice(RoomId,date\_out,Toal\_fee) |
| 交叉引用 | 酒店前台-从控机 |
| 前置条件 | 客户退房 |
| 后置条件 | 输出账单文件 |

（2）、交互图



（3）、交互图说明及层次化设计

用户界面层：与前台人员进行交互，用于发出打印账单请求。

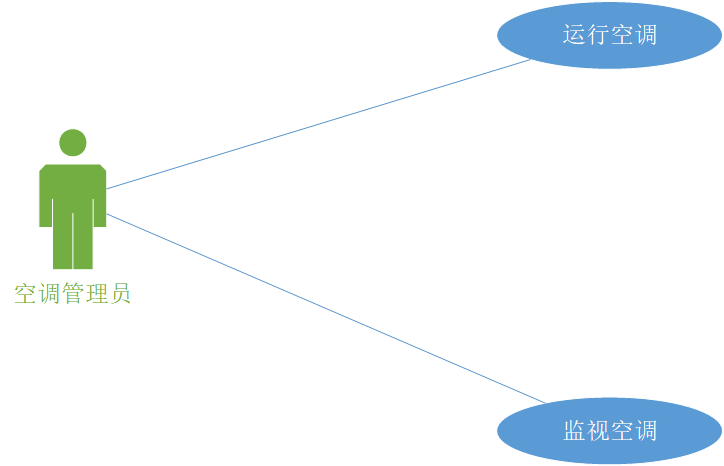
控制层：从控机作为控制层接收请求，发出创建文件以及打印账单请求

业务逻辑层：print，用于实现创建打印账单的具体功能需求，包括打印账单单和记录打印账单的日志

持久化层：ListPrint\_Log，在打印操作完成后，将打印账单操作的日志信息永久保存于数据库中

# 6. 动态结构设计——管理员

## 用例图



## 用例实现过程

### 1.PowerON()

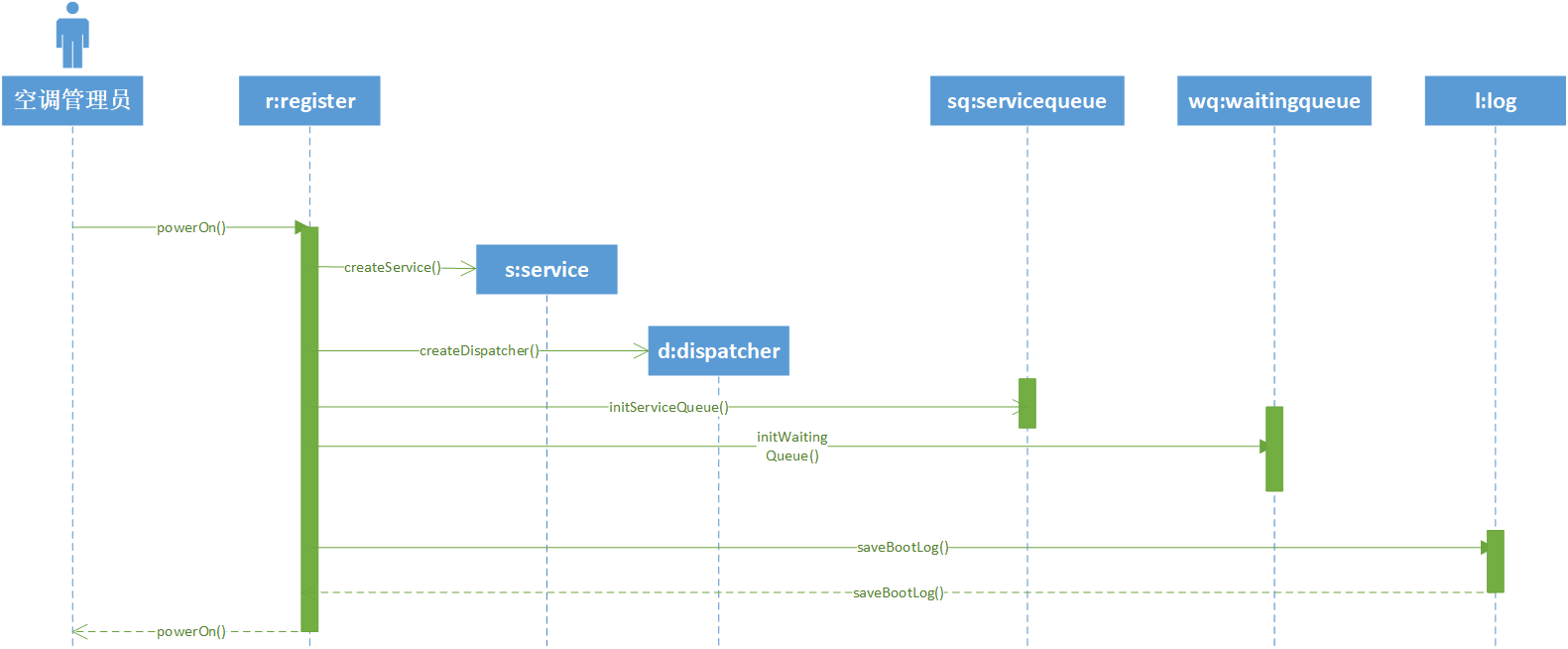
（1）对应的操作契约

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；

（2）交互过程



用户界面层：与空调管理员进行交互，用于发出启动空调请求。

控制层：register作为控制层接收请求，检查请求信息是否合法，而后发出创建对象以及队列初始化 请求。

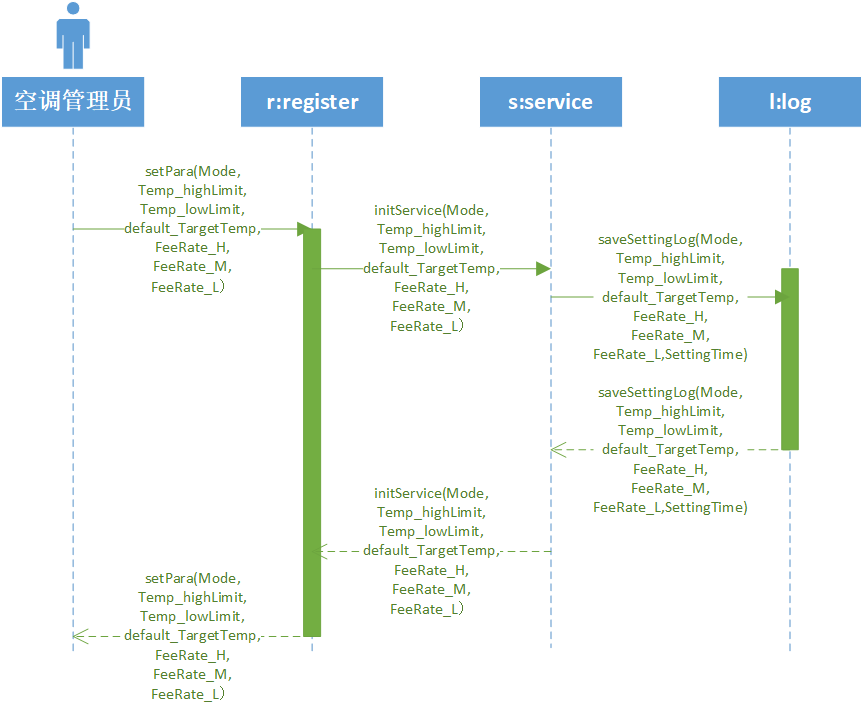
持久化层：log作为持久化层，在操作完成后，将开机的日志信息永久保存于数据库中。

### 2.setPara(Mode,Temp\_highLimit,Temp\_lowLimit,default\_TargetTemp,FeeRate\_H,FeeRate\_M,FeeRate\_L）

（1）操作契约

1、服务对象的属性初始化

（2）交互过程



用户界面层：与空调管理员进行交互，用于发出设置中央空调工作参数的请求。

控制层：register作为控制层接收请求，检查请求信息是否合法，而后发出服务对象初始化请求。

业务逻辑层：service，用于实现设置参数的具体功能需求，包括记录设置参数日志。

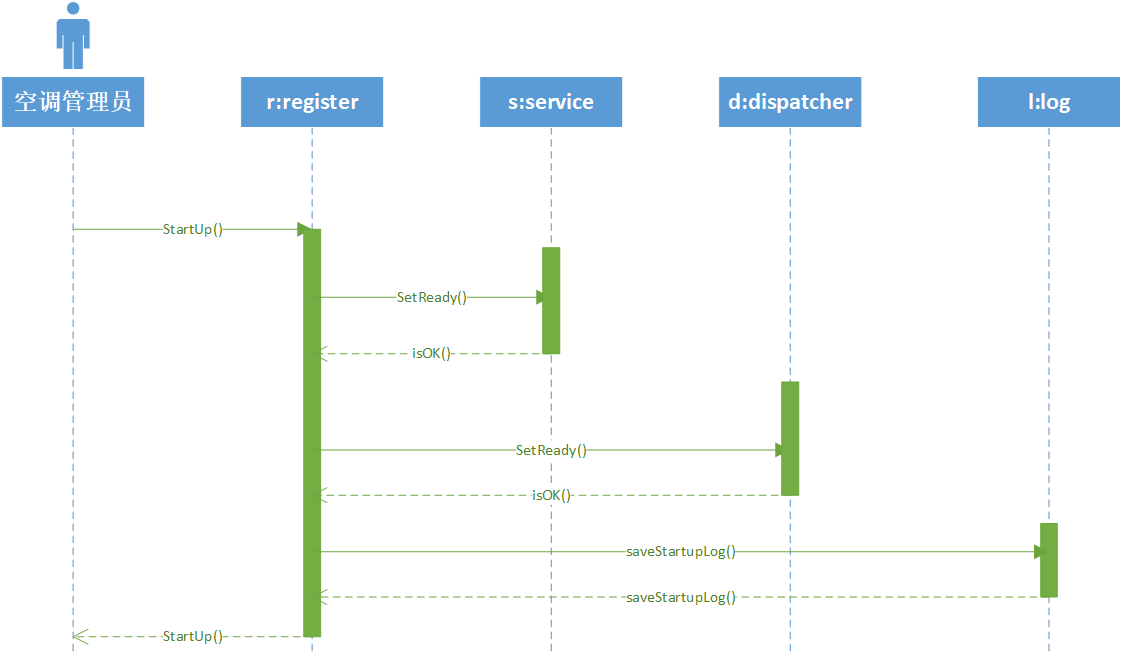
持久化层：log作为持久化层，将设置参数的日志信息永久保存于数据库中。

### 3.StartUp()

（1）操作契约

1、服务对象及调度对象的状态被设置为：就绪

（2）交互过程



用户界面层：与空调管理员进行交互，用于发出开启空调服务的请求。

控制层：register作为控制层接收请求，检查请求信息是否合法，而后发出服务对象和调度对象就绪请 求。

持久化层：log作为持久化层，在操作完成后，将开启服务的日志信息永久保存于数据库中。

### 4.CheckRoomState(list\_Room)

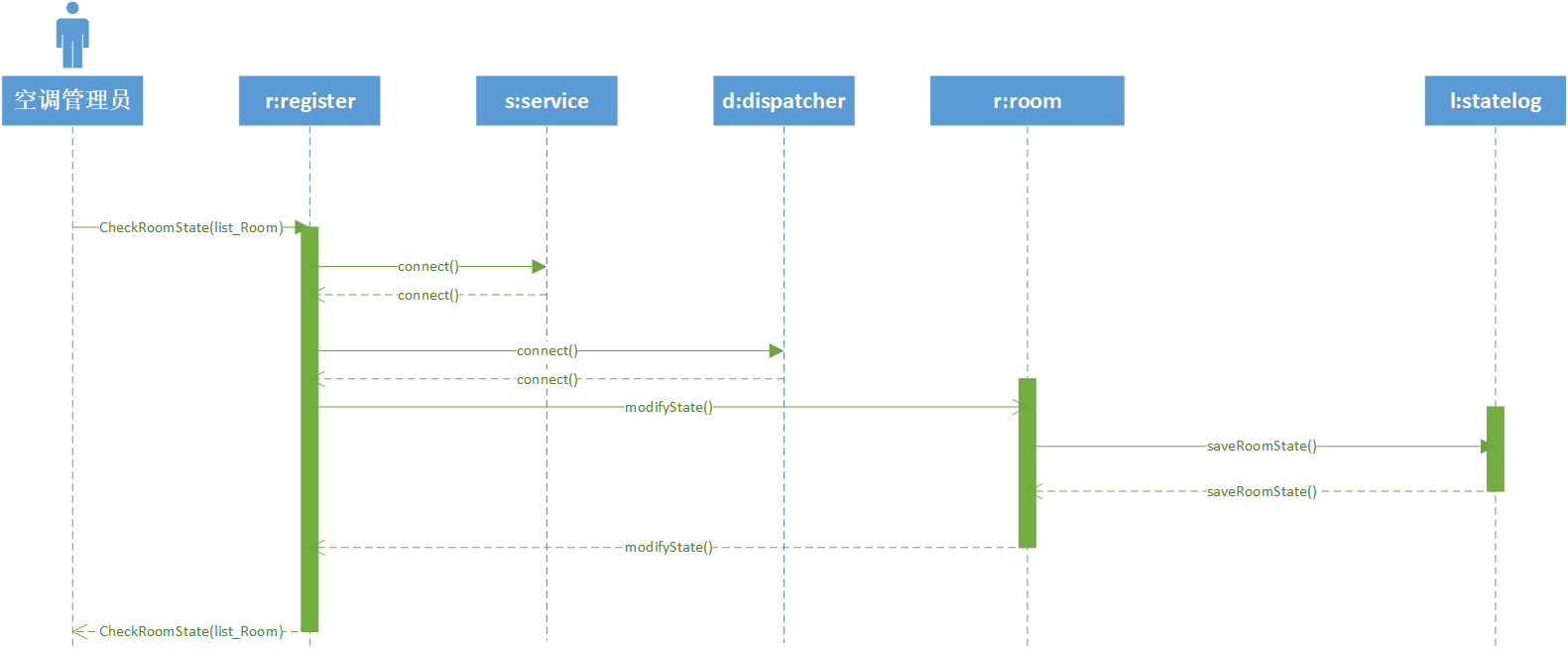
（1）操作契约

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；

（2）交互过程



用户界面层：与空调管理员进行交互，用于发出监视房间空调的请求。

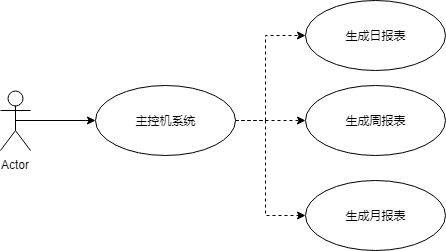
控制层：register作为控制层接收请求，检查请求信息是否合法，而后发出与服务对象和调度对象的关 联请求和修改房间空调状态的请求。

业务逻辑层：room作为业务逻辑层，用于实现修改房间空调状态的具体功能需求，包括记录修改状态 日志。

持久化层：statelog作为持久化层，在操作完成后，将修改状态的日志信息永久保存于数据库中。

# 7. 动态结构设计——经理

## 用例图



## 用例实现过程

### 1.QueryReport(list\_RoomId,type\_Report,date)

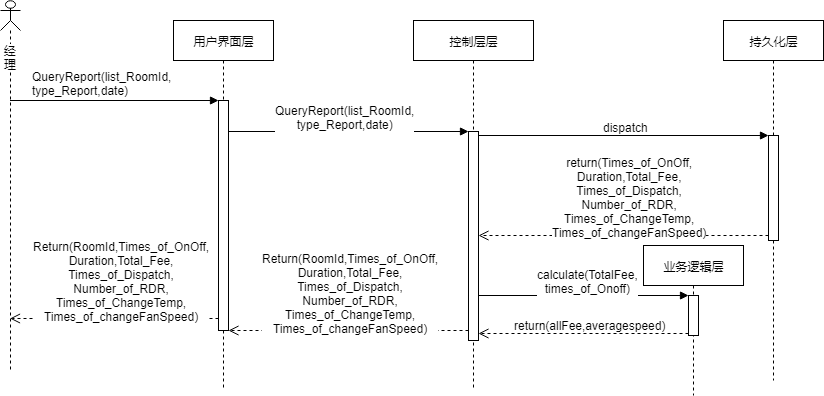
（1）操作契约

1、创建服务对象实例（或者考虑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）交互图



（3）交互图层次说明

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）。

用户界面层：有经理来使用，查询报表信息

控制层：请求被发给主控机，由主控机再去调用数据库

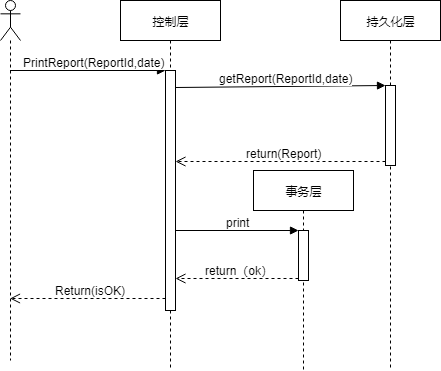
持久化层：由主控机发送的ID与pwd查找对应的用户信息，检查是否正确并返回用户的相关信息。

### 2. PrintReport(ReportId,date)

1. 操作契约

1、报表文件被创建；

（2）交互图



（3）交互图层次说明

打印报表等同于将报表内容格式化后保存于（或者导出）某种类型的文档用于离线查看。

调用持久化层取得报表数据并打印。

# 8. 动态结构设计——调度交互过程

## 调度策略

调度对象负责接收空调客户端的请求，然后转发给服务对象进行温度计算和计费；

调度对象（可以认为是控制器对象）在系统服务器启动时进行实例化和初始化，需要创建两个队列：

1、服务队列：记录房间号和服务对象的Id，风速以及服务时长；

2、等待队列：记录房间号、风速和分配的等待服务时长。

基本的调度策略为：优先级调度+时间片调度。

优先级调度的依据是请求的风速，为此调度时首先考虑优先级策略；当所有请求的风速相同时，再进行时间片调度；

1、当服务对象数小于服务对象上限时，所有请求会被分配一个服务对象；

2、当服务对象数大于等于服务对象上限时，启动调度策略：

首先判断是否符合优先级策略：请求服务的风速和服务对象的风速的大小：

2.1、如果 判断=大于，则启动优先级调度策略，再判断有几个服务对象的风速低于请求风速：

2.1.1 如果只有一个，则该房间被放置于等待队列，并被分配一个等待服务时长；服务对象被释放并被分配给新的请求对象；

2.1.2 如果有多个服务对象的风速相等且低于请求对象，则服务时长最大的服务对象被释放并分配给新的请求对象，该房间被放置于等待队列，且分配一个等待服务时长；

2.1.3 如果多个服务对象的风速低于请求风速，且风速不相等，则将风速最低的服务对象释放，该房间被放置于等待队列，且分配一个等待服务时长；

2.2 如果 判断 = 相等，则启动时间片调度策略

2.2.1 将请求对象放置于等待队列，并分配一个等待服务时长；

2.2.2 在这两分钟期间，如果没有任何服务状态的变化，当等待服务时长=0时

将服务队列中服务时长最大的服务对象释放，该房间被放置于等待队列，且被分配一个等待服务时长；

等待服务对象被分配一个服务对象开始服务；

2.2.3 在这等待的两分钟期间内，如果有任何一个服务对象的目标温度到达或关机（意味着服务对象释放）

则等待队列中的等待服务时长最小的对象获得服务对象；

2.3 如果 判断 = 小于，则 请求对象必须等到某个服务对象空闲后才能得到服务。

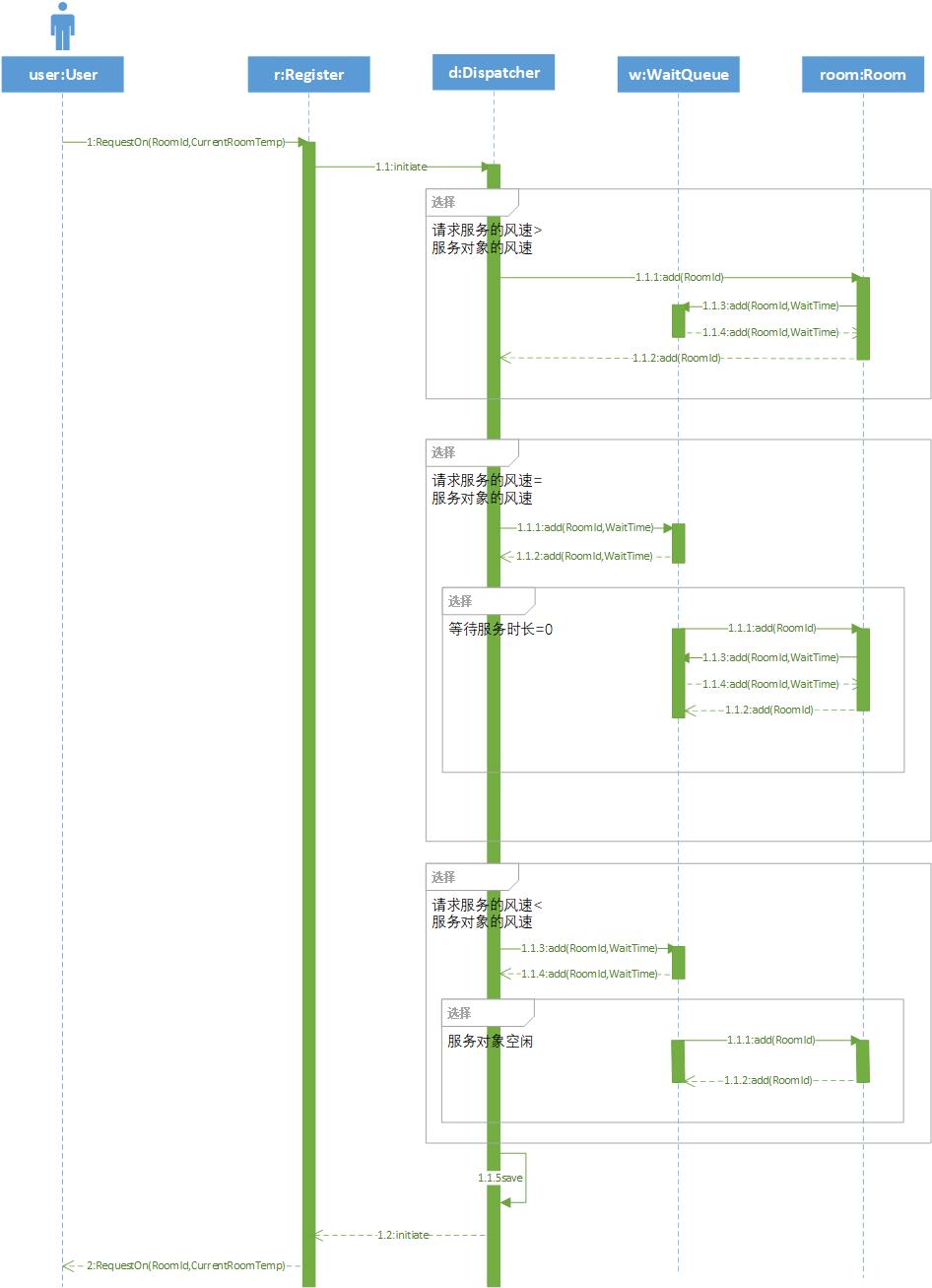
## 调度过程实现

### 1.RequestNumber>ServiceNumber

服务对象数到达上限操作契约：

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。

调度过程的交互图：



实现过程：

1. Register作为控制器对象接收系统事件RequestOn。
2. Register将服务对象数及服务开始时间赋值给调度对象Dispatcher的实例d。
3. Dispatcher根据调度策略对每个RequestOn请求判断优先级和时间片，Dispatcher根据优先级和时间片将Roomid分配到等待队列或者执行队列。
4. 关联Dispatcher和Room。
5. Dispatcher将房间请求放入等待队列进行调度。
6. 保存调度对象。