

## 分布式温控系统

# 软件项目总结报告

314c: -AirCondition-CLOSUREREPORT

版本: 1.2.1

作者:	314 班软件工程 C 组	日期:	2021-06-30

## 变更记录

日期	版本	变更说明	作者
2021-05-10	1.0.0	创建 AirCondition 项目，完成酒店前台办理入住功能	314 班 C 组
2021-05-12	1.0.1	完成中央空调基本功能，包括基础请求响应，送风	314 班 C 组
2021-05-14	1.0.2	完成基础所有功能	314 班 C 组
2021-05-16	1.1.0	修改调度策略，送风请求策略	314 班 C 组
2021-05-20	1.1.1	完成计费功能，账单查看功能	314 班 C 组
2021-06-04	1.2.0	修改前端界面，将项目部署到服务器上运行	314 班 C 组
2021-06-07	1.2.1	将前端页面完全分离并进行联调	314 班 C 组

## 1 项目信息

项目名称	分布式温控系统	项目编号	314c
项目来源	《软件工程》课程	提交时间	2021-06-07

## 2 项目说明

假定，某快捷廉价酒店响应节能环保理念，推行自助式房间温度调节的空调系统，经过初步分析该系统的需求如下：

1. 空调系统由中央空调和房间空调两部分构成；
2. 中央空调是冷暖两用，根据季节进行工作模式调整。a) 当设置为供暖时，供暖温度控制在  $25^{\circ}\text{C} \sim 30^{\circ}\text{C}$  之间； b) 当设置为制冷时，制冷温度控制在  $18^{\circ}\text{C} \sim 25^{\circ}\text{C}$  之间。
3. 中央空调具备开关按钮，只可人工开启和关闭，中央空调正常开启后处于待机状态。a) 中央空调开机后，默认处于制冷模式，缺省工作温度为  $22^{\circ}\text{C}$ ，当切换到供暖模式时，缺省工作温度为  $28^{\circ}\text{C}$ ； b) 当关闭后，不响应来自房间的任何温控请求； c) 当有来自从控机的温控要求时，中央空调开始工作； d) 当所有房间都没有温控要求时，中央空调的状态回到待机状态。
4. 房间内有独立的从控空调机，但没有冷暖控制设备。a) 从控机具有一个温度传感器，实时监测房间的温度，并与从控机的目标设置温度进行对比，从而向中央空调机发出送风请求进行温度调节。b) 如果从控机发出的请求和中央空调设置的冷暖控制状态发生矛盾时，以中央空调机的状态优先，否则中央空

调机不予响应。

5. 从控机只能人工方式开闭, 并通过控制面板设置目标温度, 目标温度有上下限制。 a) 从控机开机后动态获取房间温度, 并将温度显示在控制面板上; b) 从控机开机后需要与中央空调进行连接认证, 用户输入房间号+身份证号后, 从控机从中央空调获取工作模式和缺省工作温度, 并将它们显示在控制面板上;

6. 控制面板的温度调节可以连续变化也可以断续变化: a) 温度调节按钮连续两次或多次指令的时间间隔小于 1s 时, 从控机只发送最后一次 的指令参数; b) 如果温度调节按钮连续两次的时间间隔大于 1s 时, 从控机将发送两次指令参数;

7. 房间目标温度达到后, 从控机自动停止工作, 同时发送停止送风请求给中央空调。 a) 房间温度随着环境温度开始变化, 当房间温度超过目标温度 1 °C 时, 重新启动; b) 房间不考虑大小和管道的分布及大小问题, 在达到目标温度后, 房间温度每分钟上 下变化 X°C (各小组自行定义环境温度的变化曲线)。

8. 中央空调能够实时监测各房间的温度和状态, 并要求实时刷新的频率能够进行配置;

9. 要求从控机的控制面板能够发送高、中、低风速的请求, 要求各小组自定义高、中、低 风情况下的温度变化值; 比如以中风为基准, 高速风的温度变化曲线可以提高 25%, 低速风的温度变化曲线可以降低 25%。

10. 系统中央空调部分具备计费功能: 可根据中央空调对从控机的请求时长及高中低风速的 供风量进行费用计算; a) 每分钟中速风的能量消耗为一个标准功率单位; b) 低速风的每分钟功率消耗为 0.8 标准功率; c) 高速风的每分钟功率消耗为 1.2 标准功率; d) 并假设, 每一个标准功率消耗的计费标准是 5

元。

11. 中央空调实时计算每个房间所消耗的能量以及所需支付的金额，并将对应信息发送给每个从控机进行在线显示，以便客户可以实时查看用量和金额。

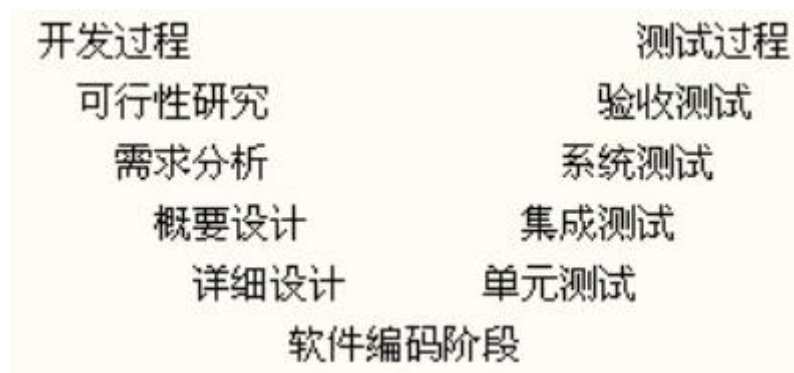
12. 中央空调监控具备统计功能，可以根据需要给出日报表、周报表和月报表；报表内容如下：房间号、从控机开关机的次数、温控请求起止时间（列出所有记录）、温控请求的起止温度及风量消耗大小（列出所有记录）、每次温控请求所需费用、每日（周、月）所需总费用。

13. 中央空调只能同时处理三台从控机的请求，为此主机要有负载均衡的能力。如果有超过三台从控机请求，则需要对所有请求机器进行调度，调度算法可自行定义，如先来先到、高速风优先抢占、时间片轮询等。

## 3 软件开发与管理过程

### 3.1 过程说明

采用的周期模型	裁剪说明
V 模型	无



### 3.2 过程改进建议

V 模型适用于需求基本清晰确认的情况，在本项目中，分布式温控系统的需求基本详细，V 模型可以很好的指导整个软件项目。需要改进的点为，由于开发时间紧，任务多，团队成员数量有限，对于软件的测试不够详细，单元测试并没有进行详细而完整的测试，改进意见为应尽可能早的考虑到软件测试，指定测试计划、编写测试用例等。

## 4 开发工具和环境

### 4.1 开发工具

- IntelliJ IDEA 2020.2.3  
编写分布式温控系统后端代码
- Microsoft Visual Studio Code  
编写分布式温控系统前端代码
- Xshell+xftp  
将项目上传至服务器并部署

### 4.2 开发环境

- Java SE Development Kit 15
- JQuery 2.1.4

## 5 估计偏差率

### 5.1 进度估计偏差率

实际进度周期	立项时估计项目周期	需求/计划时估计项目周期	立项时进度估计偏差率	需求/计划完成时进度估计偏差率
90 天	85 天	85 天	5.5%	5.5%

### 5.2 工作量估计偏差率

实际工作量	立项时估计工作量	需求/计划时估计工作量	立项时工作量差率	需求/计划完成时工作量估计偏差率
60 peron hours	50 person hours	50 person hours	20%	20%

### 5.3 代码规模估计偏差率

实际代码规模	立项时估计代码规模	需求/计划时估计代码规模	立项时代码规模估计偏差率	需求/计划完成时代码规模估计偏差率
7819 行	9000 行	9000 行	10%	10%



## 6 需求

### 6.1 需求实现情况

#### 1) 需求实现情况：

完成了《需求规格说明书》中的所有用户需求，包含中央空调与房间空调的协调工作请求、中央空调冷暖两用设计、房间空调温度传感器控制与实时显示计费、中央空调高风速优先调度策略，账单和详单查看等。

#### 2) 偏差原因说明：

基于真实情况的考虑，本系统只实现了中央空调根据温控请求风速的高低进行温控调度，未实现其他调度策略

#### 3) 调整措施：

中央空调调度策略设置为，默认进行高风速优先调度策略。

### 6.2 需求变更情况

#### 1) 需求变更情况：

中央空调需要根据季节进行冷暖风的变更，无法在中央空调正在使用时进行冷暖变更。

#### 2) 偏差原因说明：

基于真实情况的考虑，酒店中的中央空调属于大型设备，同时中央空调控制着整个酒店的温度，不可以随时更改送风的冷暖，在季节更替时进行更改是最为合理的选择。

#### 3) 调整措施：

若要更改冷暖风请求，需重启中央空调。

## 7 客户反馈

### 1) 客户反馈情况

无

### 2) 偏差原因说明

无

### 3) 调整措施

无

## 8 项目问题

### 1) 项目问题情况

项目在进行组内联调时出现 JDK 版本不一致原因，使得项目源码无法在其他小组成员电脑上运行，同时在将前端完全分离出时出现了请求失败

### 2) 偏差原因说明

JDK 版本不一致使得编译出现问题，完全分离的前端进行请求时涉及跨域问题

### 3) 调整措施

将 JDK 版本统一成 JDK 15，在后端允许跨域请求。

## 9 组内协调活动跟踪结果

协调人	协调方式	协调内容	发生问题	处理方式	频率/时间
孙兴华	腾讯会议	确认小组组长，确认需求	无	无	2021-03-08
孙兴华	腾讯会议	温控系统需求分析	分工问题	细分小组内部工作	2021-03-13
孙兴华	腾讯会议	温控系统领域模型构建	无	无	2021-04-07
孙兴华	腾讯会议	温控系统用例模型构建	无	无	2021-04-23
孙兴华	腾讯会议	温控系统动态结构设计	无	无	2021-05-20
李志毅	线下交流	温控系统代码编写与开发	无	无	2021-05-21
李志毅	线下交流	温控系统验收前测试	无	无	2021-06-06

## 10 软件项目改进建议

本次软件项目完成了《分布式温控系统》的制定计划、需求分析、软件设计、程序编写、软件测试过程，完成了所有的需求功能，通过了最终的验收测试，但该项目依旧存在着一些问题，例如：中央空调的调度策略有待改善，对于账单的计费策略也可以进行优化等等，中央空调的调度策略可以增加多种调度，由中央空调管理员进行选择策略，空调的计费也可以进行优化，根据不同的变化策略和温控范围进行计费，而不仅仅根据风速和时间来计费等等。

## 11 项目成员总结

# 项目源码清单

## 后端代码：

### SystemController.java

```
package cn.edu.bupt.aircondition.controller;

import cn.edu.bupt.aircondition.common.Result;
import cn.edu.bupt.aircondition.entity.*;
import cn.edu.bupt.aircondition.mapper.AirConditionLogMapper;
import cn.edu.bupt.aircondition.mapper.ScheduleLogMapper;
import cn.edu.bupt.aircondition.mapper.ServeMapper;
import cn.edu.bupt.aircondition.entity.Env;
import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;
import com.sun.tools.jconsole.JConsoleContext;
import io.swagger.annotations.Api;
import io.swagger.annotations.ApiOperation;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.format.annotation.DateTimeFormat;
import org.springframework.web.bind.annotation.*;

import java.math.BigDecimal;
import java.util.*;
import static cn.edu.bupt.aircondition.entity.Constant.*;

/**
 * @about a restful controller , return json data
 *
 */

/**
 * 基本步骤
 * 0、空调管理员中央空调参数配置
 * 1、前台用户入住登记
 * 2、客户空调测试
 * 3、客户退房，出详单
 * 4、酒店经理可在任何时间查询报表
 */
//todo: 设置的系统时间与本时区时间不符
//todo: 默认调度时的请求各个参数都为合法数据， 在此之前凡是需要进入调度的请求必须检查数据的合法性
@RestController
@RequestMapping
@Api(value = "中央空调系统接口测试")

public class SystemController {

    private ServeQueue serveQueue;

    private WaitQueue waitQueue;
```

```

private AirConditionList airConditionList;

private List<Customer> customerList;

@Autowired
private ServeMapper serveMapper;

@Autowired
private ScheduleLogMapper scheduleLogMapper;

@Autowired
private AirConditionLogMapper airConditionLogMapper;

private Integer state = STATE_OFF;//标识中央空调的当前状态：STATE_ON,
STATE_RUNNING, STATE_OFF

private List<Integer> reachTarget;//标识空调是否达到目标温度

/* *****客户相关
***** */

/**
 * @about: 接收空调开机请求，处理开机请求
 */
@ApiOperation(value = "开机请求")
@GetMapping("/ac/start")
public Result<?> start(@RequestParam Integer acId, @RequestParam Double
cTemp) {

    /*不存在开机请求、关机请求，开机时要设置空调属性，然后发送默认的送风请求*/

    if (!state.equals(STATE_RUNNING)){
        return Result.error(500, "系统未运行!!");
    }

    /*不检查客户是否存在，因为空调管理员可以控制空调*/

    /*设置空调初始化参数*/
    /*fee、scheduleTime、useTime属性在客户入住时设置*/
    AirCondition airCondition=airConditionList.getAirCondition(acId);

    //判断空调是否已经开机
    if (!airCondition.getState().equals(STATE_OFF)){
        return Result.error(500, "空调已开机");
    }

    //设置默认开机状态
    airCondition.setState(STATE_ON);
    airCondition.setFee(new BigDecimal("0.0"));
    airCondition.setTargetTemperature(TEMP_DEFAULT);
    airCondition.setSpeed(SPEED_DEFAULT);
    //写入开机记录
    //airConditionLogMapper.insert(new AirConditionLog(acId, STATE_ON,
systemClock));

    //设置初始温度

```

```

        airCondition.setOriginTemperature(new
BigDecimal(Double.toString(cTempt)));
        //设置当前温度
        airCondition.setCurrentTemperature(new
BigDecimal(Double.toString(cTempt)));
        List<AirCondition>
airConditionList1=airConditionList.getAirConditionList();
        airConditionList1.set(acId,airCondition);
        airConditionList.setAirConditionList(airConditionList1);//修改
        //设置空调到达目标温度位
        reachTarget.set(airCondition.getAcId(), NO_REACH_TARGET);
        /*封装并发送请求*/
        waitObject waitObject = new waitObject(acId, SPEED_DEFAULT, 0);
        schedule(waitObject);
        return Result.ok(waitObject);
    }

    /**
     * @about: 接收空调关机请求，处理关机请求
     */
    @ApiOperation(value = "关机请求")
    @GetMapping("/ac/shutdown")
    public Result<?> shutdown(@RequestParam Integer acId) {

        AirCondition airCondition = airConditionList.getAirCondition(acId);

        if (!state.equals(STATE_RUNNING)){
            return Result.error(500, "系统未运行!!");
        }
        //检查空调是否已经关机
        if (airCondition.getState().equals(STATE_OFF)){
            return Result.error(500,"空调已经关机");
        }
        /*检查ServeQueue中是否有请求；如果有，设置state为0，即空闲，同时将数据写入数据库*/
        for (ServeObject value : serveQueue.getServeObjectList()) {
            if (value.getState().equals(SERVE_RUNNING) &&
value.getAcId().equals(acId)) {
                value.setEndTime(systemClock);//设置退出服务队列的时间
                //存入数据库
                serveMapper.insert(value);
                value.setState(SERVE_IDLE);
                break;
            }
        }

        /*检查waitQueue中是否有请求；如果有，删除该请求*/
        waitQueue.delete(acId);

        /*设置空调state*/
        //AirCondition
        airCondition=airConditionList.getAirConditionList().get(acId);

        airCondition.setState(STATE_OFF);

        //写入关机记录

```

```

        airConditionLogMapper.insert(new AirConditionLog(acId, STATE_OFF,
systemClock));

        return Result.ok(airCondition);
    }

    /**
     * @about: 接收调节风速、温度请求，处理请求
     * dewind、deTemp 为 目标风速、目标温度
     * todo: 参数如果未被设置，前端使用当前空调的参数，
     */
    @ApiOperation(value = "调节风速、温度请求")
    @GetMapping("/ac/adjust")
    public Result<?> adjust(@RequestParam Integer acId, @RequestParam Integer
dewind, @RequestParam Double deTemp) {
        if (!state.equals(STATE_RUNNING)){
            return Result.error(500, "系统未运行!!");
        }
        /*检查空调是否开机*/
        AirCondition airCondition = airConditionList.getAirCondition(acId);
        if(airCondition.getState().equals(STATE_OFF)){
            return Result.error(500,"该空调未开机! ");
        }
        if (deTemp != -1 ){
            BigDecimal deTempB = new BigDecimal(Double.toString(deTemp));
            //判断温度是否在温度控制范围内
            if (deTempB.compareTo(TEMPT_LOW_LIMIT) < 0 ||
deTempB.compareTo(TEMPT_HIGH_LIMIT) > 0){
                return Result.error(500, "目标温度不在温度控制范围内");
            }

            airCondition.setTargetTemperature(deTempB);
        }
        if (dewind != -1){
            //          waitObject waitObject = new WaitObject(acId, dewind,
airCondition.getTargetTemperature(), 0);
            //airCondition.setSpeed(dewind);
            waitObject waitObject = new waitObject(acId, dewind, 0);
            schedule(waitObject);
        }
        return Result.ok("设置参数成功");
    }

    @ApiOperation(value = "获取单个空调状态")
    @GetMapping("/ac/state")
    public Result<?> getAcState(@RequestParam Integer acId){
        //空调状态没有更新
        return Result.ok(airConditionList.getAirCondition(acId));
    }

    /** *****空调管理员相关
    *****/

    /**
     * @about: 处理中央空调开启请求

```



```

        */
        @ApiOperation(value = "开启中央空调")
        @GetMapping("/admin/start")
        public Result<?> turnOnSystem(@RequestParam Integer
numOfServeObject,@RequestParam Integer defaultMode,@RequestParam Integer
defaultSpeed, @RequestParam Double temptHighLimit, @RequestParam Double
temptLowLimit, @RequestParam Double defaultTempt,
                                     @RequestParam Double feeRateH, @RequestParam
Double feeRateM, @RequestParam Double feeRateL ){
            if(state.equals(STATE_OFF)){
                state = STATE_RUNNING;
                serveQueue = new ServeQueue();
                waitQueue = new WaitQueue();
                airConditionList = new AirConditionList();
                customerList = new ArrayList<>();
                reachTarget = new ArrayList<>(NUM_OF_AIR_CONDITION);
                for (int i = 0;i < NUM_OF_AIR_CONDITION;i++){
                    reachTarget.add(NO_REACH_TARGET);
                }

                MODE_DEFAULT = defaultMode;
                SPEED_DEFAULT = defaultSpeed;
                TEMPT_DEFAULT = new BigDecimal(Double.toString(defaultTempt));
                TEMPT_HIGH_LIMIT = new BigDecimal(Double.toString(temptHighLimit));
                TEMPT_LOW_LIMIT = new BigDecimal(Double.toString(temptLowLimit));
                feeMap.put(SPEED_H, new BigDecimal(Double.toString(feeRateH)));
                feeMap.put(SPEED_M, new BigDecimal(Double.toString(feeRateM)));
                feeMap.put(SPEED_L, new BigDecimal(Double.toString(feeRateL)));
                temptMap.put(SPEED_L, new BigDecimal("0.4"));
                temptMap.put(SPEED_M, new BigDecimal("0.5"));
                temptMap.put(SPEED_H, new BigDecimal("0.6"));

                serveQueue.init(numOfServeObject);
                serveQueue.setDefaultPara(defaultMode, defaultSpeed,
feeMap.get(defaultSpeed),
                                TEMPT_DEFAULT, TEMPT_DEFAULT, TEMPT_DEFAULT,new
BigDecimal("0.0"));
                airConditionList.setDefaultPara(TEMPT_DEFAULT, defaultMode,
TEMPT_DEFAULT, defaultSpeed, STATE_OFF, new BigDecimal("0.0"));
                serveQueue.setReady();
                return Result.ok(serveQueue.getServeObjectList());
            }else{
                return Result.error(500,"中央空调已开机");
            }
        }

    }

    @ApiOperation(value = "获取空调运行状态")
    @GetMapping("/admin/state")
    public Result<?> getState(){
        //会在这里更新空调状态
        if (!state.equals(STATE_RUNNING)){
            return Result.error(500, "空调未运行");
        }
        //更新系统时间 点击即系统运行一分钟
        systemClock.setMinutes(systemClock.getMinutes()+1);
        //更新等待队列的等待时长

```

```

waitQueue.update();

//对于达到目标温度后未被服务的空调,每分钟温度变化0.5
for (AirCondition a:airConditionList.getAirConditionList()) {
    //处于待机状态
    if (a.getState().equals(STATE_ON) &&
reachTarget.get(a.getAcId()).equals(REACH_TARGET)){
        //温度变化0.5度
        if (a.getMode().equals(MODE_H)){
            //处于制热模式下,当前温度 > 原始温度, 不能小于原始温度

            a.setCurrentTemperature(a.getCurrentTemperature().subtract(new
BigDecimal("0.5")).compareTo(a.getOriginTemperature()) > 0?
a.getCurrentTemperature().subtract(new
BigDecimal("0.5")):a.getOriginTemperature());
        }else if (a.getMode().equals(MODE_R)){
            //处于制冷模式下,当前温度 < 原始温度, 不能大于原始温度
            a.setCurrentTemperature(a.getCurrentTemperature().add(new
BigDecimal("0.5")).compareTo(a.getOriginTemperature()) < 0?
a.getCurrentTemperature().add(new BigDecimal("0.5")):a.getOriginTemperature());
        }
        //当前温度和目标温度差大于1度
        if
(a.getCurrentTemperature().subtract(a.getTargetTemperature()).abs().compareTo(ne
w BigDecimal("1.0")) >= 0){
            reachTarget.set(a.getAcId(), NO_REACH_TARGET);
            waitObject waitObject = new waitObject();
            waitObject.setAcId(a.getAcId());
            waitObject.setSpeed(a.getSpeed());
            waitObject.setTimeToWait(TIME_TO_WAIT);
            waitQueue.insert(waitObject);
        }
    }

    //对于关机的空调,温度变化0.5度
    if (a.getState().equals(STATE_OFF)){
        //温度变化0.5度
        if (a.getMode().equals(MODE_H)){
            //处于制热模式下,当前温度 > 原始温度, 不能小于原始温度

            a.setCurrentTemperature(a.getCurrentTemperature().subtract(new
BigDecimal("0.5")).compareTo(a.getOriginTemperature()) > 0?
a.getCurrentTemperature().subtract(new
BigDecimal("0.5")):a.getOriginTemperature());
        }else if (a.getMode().equals(MODE_R)){
            //处于制冷模式下,当前温度 < 原始温度, 不能大于原始温度
            a.setCurrentTemperature(a.getCurrentTemperature().add(new
BigDecimal("0.5")).compareTo(a.getOriginTemperature()) < 0?
a.getCurrentTemperature().add(new BigDecimal("0.5")):a.getOriginTemperature());
        }
    }
}

//正在接收服务的空调
for (ServeObject s:serveQueue.getServeObjectList()) {
    if (s.getState().equals(SERVE_RUNNING)){
        //更新对应房间的状态
        AirCondition airCondition =
airConditionList.getAirCondition(s.getAcId());

```

```

        //更新温度

        BigDecimal currentTemp = s.getMode().equals(MODE_H)?
s.getCurrentTemp().add(tempMap.get(s.getSpeed())):s.getCurrentTemp().subtract(t
emptMap.get(s.getSpeed()));
        System.out.println(currentTemp);
        if (s.getMode().equals(MODE_H)){
            //最高只能达到目标温度，不能超过目标温度
            if
(currentTemp.compareTo(airCondition.getTargetTemperature()) > 0){
                s.setCurrentTemp(airCondition.getTargetTemperature());
            }else{
                s.setCurrentTemp(currentTemp);
            }
        }else if (s.getMode().equals(MODE_R)){
            //达到目标温度
            if
(currentTemp.compareTo(airCondition.getTargetTemperature()) < 0){
                s.setCurrentTemp(airCondition.getTargetTemperature());
            }else{
                s.setCurrentTemp(currentTemp);
            }
        }
        //更新服务时长
        s.setServeTime(s.getServeTime() + 1);
        //更新费用
        s.setFee(s.getFee().add(feeMap.get(s.getSpeed())));
        //空调记录总费用

        airCondition.setFee(airCondition.getFee().add(feeMap.get(s.getSpeed())));
        airCondition.setCurrentTemperature(s.getCurrentTemp());
        //判断是否达到目标温度
        if
(airCondition.getCurrentTemperature().equals(airCondition.getTargetTemperature()
)){
            //出服务队列
            System.out.println("out"+systemClock);
            s.setEndTime(systemClock);
            //记录保存到数据库
            serveMapper.insert(s);
            //服务对象空闲
            s.setState(SERVE_IDLE);

            reachTarget.set(airCondition.getAcId(), REACH_TARGET);
            //空调未被服务
            airCondition.setState(STATE_ON);
        }
    }
}
schedule();
Env env = new Env();
env.setAirConditionList(airConditionList);
env.setServeQueue(serveQueue);
env.setWaitQueue(waitQueue);
return Result.ok(env);
}

/* @about: 关闭中央空调

```

```

    */
    @ApiOperation(value = "关闭中央空调")
    @GetMapping("/admin/off")
    public Result<?> turnOff(){
        state = STATE_OFF;
        airConditionList.setOff();
        serveQueue.setReady();
        return Result.ok();
    }

    /* *****前台服务员
    ***** */

    /**
     * @about 创建账单及详单
     */
    public BillswithDetails createBillswithDetails(@RequestParam Integer acId) {
        /*查找客户入住、退房时间*/
        Customer customer = null;
        for (Customer value : customerList) { //遍历所有客户，查找acId对应的客户
            if (value.getAcId() == acId) {
                customer = value;
                break;
            }
        }
        if (customer == null) { //若不存在该客户
            return null;
        }

        long startTime = customer.getInTime().getTime();
        long endTime = customer.getOutTime().getTime();
        Date startTimeDate = new Date(startTime);
        Date endTimeDate = new Date(endTime);

        QueryWrapper<ServeObject> queryWrapper = new QueryWrapper<ServeObject>
();
        queryWrapper.eq("ac_id", acId).ge("start_time",
startTimeDate).le("end_time", endTimeDate);
        List<ServeObject> serveObjectList =
serveMapper.selectList(queryWrapper);

        /*组装详单*/
        //创建空账单及详单
        BillswithDetails billswithDetails = new BillswithDetails();
        List<DetailObject> detailObjectList = new ArrayList<>();
        BigDecimal allFee = new BigDecimal("0.0"); //总费用
        BigDecimal allConsumption = new BigDecimal("0.0"); //总耗电

        //组装详单
        for (ServeObject value : serveObjectList) {
            DetailObject detailObject = new DetailObject();

            /*账单信息*/
            allConsumption = allConsumption.add(value.getFee());
            allFee = allFee.add(value.getFee());

            detailObject.setAcId(value.getAcId());

```

```

        detailObject.setSpeed(value.getSpeed());
        detailObject.setMode(value.getMode());
        detailObject.setStartTime(value.getStartTime());
        detailObject.setEndTime(value.getEndTime());
        detailObject.setStartTemp(value.getStartTemp()); //设置开始温度
        detailObject.setEndTemp(value.getCurrentTemp());
        detailObject.setConsumption(value.getFee());
        detailObject.setFee(value.getFee());

        /*添加详单项*/
        detailObjectList.add(detailObject);
    }

    /*添加详单列表*/
    billswithDetails.setDetailObjectList(detailObjectList);

    /*添加账单项*/
    billswithDetails.setCustomerId(customer.getCustomerId());
    billswithDetails.setName(customer.getName());
    billswithDetails.setAcId(customer.getAcId());
    billswithDetails.setStartTime(customer.getInTime());
    billswithDetails.setEndTime(customer.getOutTime());
    billswithDetails.setAllConsumption(allConsumption);
    billswithDetails.setAllFee(allFee);

    return billswithDetails;
}

/**
 * @about 客户入住
 */
@ApiOperation(value = "客户入住")
@GetMapping("/front/checkin")
public Result<?> checkIn(@RequestParam String name) {

    /*判断是否有空房*/
    Integer freeNum = -1; //值为id最小的空房
    Boolean[] is_free = new Boolean[NUM_OF_AIR_CONDITION]; //用来检查对应id的
    房间是否空闲
    for (int i = 0; i < NUM_OF_AIR_CONDITION; i++) { //初始化
        is_free[i] = true;
    }
    for (Customer value : customerList) { //遍历所有房间，检查是否空闲
        is_free[value.getAcId()] = false;
    }

    for (int i = 0; i < NUM_OF_AIR_CONDITION; i++) { //找到最小id的空房
        if (is_free[i]) {
            freeNum = i;
            break;
        }
    }

    if (freeNum == -1) {
        return Result.error("无空房!");
    }

    /*创建客户，给客户分配房间，初始化客户信息*/

```

```

//设置customer的id
Integer cusId=0;
for(Customer value:customerList){
    if(value.getCustomerId()>=cusId){
        cusId=value.getCustomerId()+1;
    }
}

/*创建客户，分配房间，并初始化信息*/
Date date=new Date(systemClock.getTime()-1000);
Customer customer = new Customer();
customer.setAcId(cusId);
customer.setInTime(date);
customer.setOutTime(date);
customer.setName(name);
customer.setAcId(freeNum);

customerList.add(customer);

/*初始化空调的fee、scheduleTime、useTime*/
AirCondition airCondition =
airConditionList.getAirConditionList().get(freeNum);
List<AirCondition>
airConditionList1=airConditionList.getAirConditionList();

airCondition.setFee(new BigDecimal("0.0"));
airConditionList1.set(freeNum, airCondition);
airConditionList.setAirConditionList(airConditionList1);

return Result.ok(customer);
}

/**
 * @about 客户退房
 */
@ApiOperation(value = "客户退房")
@GetMapping("/front/checkout")
public Result<?> checkOut(@RequestParam Integer acId) {
    Boolean flag=true; //标识客户是否存在
    for (Customer value : customerList) {
        if (value.getAcId() == acId) {
            // value.setOutTime(systemClock);
            flag=false;
            break;
        }
    }

    if(flag)
        return Result.error("客户不存在!");

    /*检查ServeQueue中是否有请求；如果有，设置state为0，即空闲，同时将数据写入数据库*/
    for (ServeObject value : serveQueue.getServeObjectList()) {
        if (value.getAcId() == acId && value.getState() != SERVE_IDLE) { //重
大的bug

            //退出服务队列之前，设置属性
            value.setEndTime(systemClock);//设置退出服务队列的时间

            //计算结束时温度、耗电量、费用

```

```

        BigDecimal duration = new
BigDecimal(Double.toString((value.getEndTime().getTime() -
value.getStartTime().getTime()) * 1.0 / 60000));
        BigDecimal fee = duration.multiply(new
BigDecimal(value.getFeeRate().toString())); //入住时长
        BigDecimal endTempt=new
BigDecimal(value.getStartTemp().toString()); //结束温度

        if (value.getSpeed().equals(SPEED_L)) {
            duration=duration.multiply(new
BigDecimal(value.getMode().equals(MODE_H)?"0.4":"-0.4")); //转换时间为温度差
            endTempt=endTempt.add(duration);
        } else if (value.getSpeed().equals(SPEED_M)) {
            duration=duration.multiply(new
BigDecimal(value.getMode().equals(MODE_H)?"0.5":"-0.5"));
            endTempt = endTempt.add(duration);
        } else {
            duration=duration.multiply(new
BigDecimal(value.getMode().equals(MODE_H)?"0.6":"-0.6"));
            endTempt = endTempt.add(duration);
        }

        value.setFee(fee);
        value.setCurrentTemp(endTempt);

        //存入数据库
        serveMapper.insert(value);

        //设置ServeObject的状态
        List<ServeObject>
serveObjectList1=serveQueue.getServeObjectList();
        value.setState(SERVE_IDLE);
        serveObjectList1.set(value.getServeId(),value);
        break;
    }
}

/*检查waitQueue中是否有请求；如果有，删除该请求*/
waitQueue.delete(acId);

/*设置客户退房时间*/
for (Customer value : customerList) {
    if (value.getAcId() == acId) {
        value.setOutTime(new Date(SystemClock.getTime()+2000));
        break;
    }
}

/*设置空调state*/
AirCondition airCondition =
airConditionList.getAirConditionList().get(acId);
    List<AirCondition>
airConditionList1=airConditionList.getAirConditionList();

    airCondition.setState(STATE_OFF);

    airConditionList1.set(acId, airCondition);

```

```

        airConditionList.setAirConditionList(airConditionList1);

        /*打印账单、详单*/
        BillswithDetails billswithDetails = createBillswithDetails(acId);

        /*删除客户*/
        for (Customer value : customerList) {
            if (value.getAcId() == acId) {
                customerList.remove(value);
                break;
            }
        }

        return Result.ok(billswithDetails);
    }

    /** *****酒店经理
    ***** */
    /**
     * 调度次数计算:
     *     特殊情况: 某个空调的旧请求被新请求替换, 调度次数算1次, 只计算了调度入队列
     *     空调关机也未被计算在内
     *
     * */

    /**
     *
     * @param type 报表类型: REPORT_DAY 日报表, REPORT_WEEK 周报表, REPORT_MONTH 月
    报表, REPORT_YEAR 年报表
     * @param date 报表开始日期
     *
     * */
    @ApiOperation(value = "获取报表")
    @GetMapping("/manager/report")
    public Result<?> getReport(@RequestParam Integer type, @RequestParam
    @DateTimeFormat(pattern="yyyy-MM-dd") Date date){

        Date start_time = null;
        Date end_time = null;
        if (type.equals(REPORT_YEAR)){
            start_time = new Date(date.getYear(), 0, 0);
            end_time = new Date(date.getYear()+1, 0, 0);
        }else if (type.equals(REPORT_MONTH)){
            start_time = new Date(date.getYear(), date.getMonth(),1);
            end_time = new Date(date.getYear(), date.getMonth()+1,1);
        }else if (type.equals(REPORT_WEEK)){
            start_time = new Date(date.getYear(), date.getMonth(),
            date.getDate());
            end_time = new Date(date.getYear(), date.getMonth(),
            date.getDate()+7);
        }else{
            start_time = date;
            end_time = new Date(date.getYear(), date.getMonth(),
            date.getDate()+1);

```



```

    }
    System.out.println(date);
    System.out.println(start_time);
    System.out.println(end_time);
    Report report = new Report();
    report.setStart_time(start_time);
    report.setEnd_time(end_time);
    List<ReportItem> reportItems = report.getReportItemList();
    for (AirCondition a:airConditionList.getAirConditionList()) {
        ReportItem reportItem = new ReportItem();
        reportItem.setAcId(a.getAcId());
        //获取使用空调次数
        reportItem.setUseTimes(airConditionLogMapper.selectCount(new
QueryWrapper<AirConditionLog>().ge("date", start_time)
        .le("date", end_time).eq("ac_id", a.getAcId()))/2);
        //最常用目标温度
        reportItem.setCommonTargetTemperature(serveMapper.getCommonTargetTemp(a.getAcId()
        (),start_time,end_time));
        //最常用风速
        reportItem.setCommonSpeed(serveMapper.getCommonSpeed(a.getAcId(),start_time,end
        _time));
        //到达目标温度次数
        reportItem.setReachTimes(serveMapper.getReachTargetTimes(a.getAcId(),start_time
        ,end_time));
        //被调度次数
        reportItem.setScheduleTimes(scheduleLogMapper.selectCount(new
QueryWrapper<ScheduleLog>().ge("date", start_time)
        .le("date", end_time).eq("ac_id", a.getAcId())));
        //详单记录数
        reportItem.setDetailRecordTimes(serveMapper.getDetailCount(a.getAcId(),start_ti
        me,end_time));
        //总费用
        reportItem.setCost(serveMapper.getFee(a.getAcId(),start_time,end_time));

        reportItems.add(reportItem);
    }
    return Result.ok(report);
}
@GetMapping("/getDetailList")
public Result<?> getDetailList(@RequestParam Integer acId,@RequestParam
Integer type, @RequestParam @DateTimeFormat(pattern="yyyy-MM-dd") Date date){
    Date start_time = null;
    Date end_time = null;
    if (type.equals(REPORT_YEAR)){
        start_time = new Date(date.getYear(), 0, 0);
        end_time = new Date(date.getYear()+1, 0, 0);
    }else if (type.equals(REPORT_MONTH)){
        start_time = new Date(date.getYear(), date.getMonth(),1);
        end_time = new Date(date.getYear(), date.getMonth()+1,1);
    }else if(type.equals(REPORT_WEEK)){
        start_time = new Date(date.getYear(), date.getMonth(),
date.getDate());

```

```

        end_time = new Date(date.getYear(), date.getMonth(),
date.getDate()+7);
    }else{
        start_time = date;
        end_time = new Date(date.getYear(), date.getMonth(),
date.getDate()+1);
    }
    QueryWrapper<ServeObject> queryWrapper = new QueryWrapper<ServeObject>
();
    queryWrapper.eq("ac_id", acId).ge("start_time",
start_time).le("end_time", end_time);
    List<ServeObject> serveObjectList =
serveMapper.selectList(queryWrapper);
    BillswithDetails billswithDetails = new BillswithDetails();
    List<DetailObject> detailObjectList = new ArrayList<>();
    BigDecimal allFee = new BigDecimal("0.0");           //总费用
    BigDecimal allConsumption = new BigDecimal("0.0");    //总耗电

    //组装详单
    for (ServeObject value : serveObjectList) {
        DetailObject detailObject = new DetailObject();

        /*账单信息*/
        allConsumption = allConsumption.add(value.getFee());
        allFee = allFee.add(value.getFee());

        detailObject.setAcId(value.getAcId());
        detailObject.setSpeed(value.getSpeed());
        detailObject.setMode(value.getMode());
        detailObject.setStartTime(value.getStartTime());
        detailObject.setEndTime(value.getEndTime());
        detailObject.setStartTemp(value.getStartTemp()); //设置开始温度
        detailObject.setEndTemp(value.getCurrentTemp());
        detailObject.setConsumption(value.getFee());
        detailObject.setFee(value.getFee());

        /*添加详单项*/
        detailObjectList.add(detailObject);
    }

    /*添加详单列表*/
    billswithDetails.setDetailObjectList(detailObjectList);

    /*添加账单项*/

    billswithDetails.setStartTime(start_time);
    billswithDetails.setEndTime(end_time);
    billswithDetails.setAllConsumption(allConsumption);
    billswithDetails.setAllFee(allFee);

    return Result.ok(billswithDetails);
}

```

```

/* *****调度
***** */

/**
 * 有空闲服务对象和从等待队列中调度处的等待对象
 * @param serveId 服务对象Id not null
 * @param waitObject 等待对象 not null
 */
private void join(Integer serveId, waitObject waitObject){
    AirCondition airCondition =
airConditionList.getAirCondition(waitObject.getAcId());
    //检查空调是否被服务,未被服务, 返回null;被服务,返回服务对象
    ServeObject oldServeObject = serveQueue.check(waitObject.getAcId());
    if(oldServeObject != null){//丢弃旧服务
        oldServeObject.setEndTime(systemClock);
        serveMapper.insert(oldServeObject);
        oldServeObject.setState(SERVE_IDLE);
    }
    ServeObject serveObject = serveQueue.getServeObject(serveId);
    //设置新分配的服务对象的状态
    serveObject.setAcId(waitObject.getAcId());
    serveObject.setState(SERVE_RUNNING);
    System.out.println(systemClock);
    System.out.println(systemClock.getTime());
    serveObject.setStartTime(new Date(systemClock.getTime()));
    serveObject.setMode(getMode(airCondition.getTargetTemperature()));
    serveObject.setEndTime(null);
    serveObject.setStartTemp(airCondition.getCurrentTemperature());
    serveObject.setCurrentTemp(airCondition.getCurrentTemperature());
    serveObject.setSpeed(waitObject.getSpeed());
    serveObject.setFeeRate(feeMap.get(waitObject.getSpeed()));
    serveObject.setFee(new BigDecimal("0.0"));
    serveObject.setServeTime(0);
    serveObject.setTargetTemp(airCondition.getTargetTemperature());
    //设置空调的状态
    airCondition.setState(STATE_RUNNING);
    airCondition.setMode(serveObject.getMode());
    airCondition.setSpeed(serveObject.getSpeed());
//    airCondition.setTargetTemperature(waitObject.getTargetTemp());
}

/**
 * 当有新的请求到来时,处理新请求
 * @param waitObject 等待对象 not null
 *
 */
//更新: 空调正在服务, 旧服务对象直接替换新服务对象

/*
 * 新请求到来时的调度规则:
 * 1、删除等待队列中已有的旧请求
 * 2、若服务队列有空闲服务对象, 将等待对象直接加入服务队列

```

```

        * 3、如果服务队列没有空闲服务对象，比较此等待对象和服务队列中所有服务对象的优先级，若比服务
        队列中某些服务对象优先级高，转4，；若和某些服务对象优先级相等，转5；比所有服务对象优先级低，转
        6

        * 4、用此等待对象替换服务队列中优先级低的服务对象，并将其加入到等待队列中，具体替换规则，
        转7

        * 5、直接插入等待队列，分配默认时间片
        * 6、直接插入等待队列，分配默认时间片
        * 7、选出所有优先级低于该等待对象的服务队列、从最低优先级的服务对象中选择出服务时长最长的
        服务对象替换
        */

    /*
    * 针对正在接收服务的空调的新请求的调度规则
    * 1、新请求比较优先级比服务队列中的某些服务对象优先级高，用新请求替换旧请求
    * 2、新请求置于等待队列中，等待调度，旧请求仍旧服务
    */

    private void schedule(waitObject waitObject){
        //如果存在,删除等待队列中的旧请求
        waitQueue.delete(waitObject.getAcId());
        //有空闲的服务对象
        Integer serveId = serveQueue.hasIdle();
        if (serveId != -1){
            join(serveId, waitObject);
        }
        //需要调度
        else{
            //比较优先级
            Integer s = serveQueue.comparePriority(waitObject.getSpeed());
            //比服务对象优先级小
            if (s.equals(-1) || s.equals(0)){
                waitObject.setTimeToWait(TIME_TO_WAIT);
                waitQueue.insert(waitObject);
            }
            //与服务对象优先级相等
            // if(s.equals(0)){
            //     //设置等待时长
            //     waitObject.setTimeToWait(TIME_TO_WAIT);
            //     waitQueue.insert(waitObject);
            // }
            //比服务对象优先级高
            if (s.equals(1)){
                AirCondition airCondition =
                airConditionList.getAirCondition(waitObject.getAcId());
                //保存中风速的服务对象
                List<ServeObject> serveObjectM = new ArrayList<>();
                //保存低风速的服务对象
                List<ServeObject> serveObjectL = new ArrayList<>();
                for (ServeObject s1: serveQueue.getServeObjectList()) {
                    //如果旧请求在被服务，新请求被调度到服务队列
                    if (s1.getAcId().equals(waitObject.getAcId())) {
                        //旧请求提前结束，写入数据库
                        s1.setEndTime(SystemClock);
                        serveMapper.insert(s1);
                        //新请求替换旧请求所在服务对象
                        s1.setServeTime(0);
                        s1.setFee(new BigDecimal("0.0"));
                        s1.setFeeRate(feeMap.get(waitObject.getSpeed()));
                        s1.setTargetTemp(airCondition.getTargetTemperature());
                    }
                }
            }
        }
    }
}

```

```

        s1.setSpeed(waitObject.getSpeed());
        s1.setStartTemp(airCondition.getCurrentTemperature());
        s1.setCurrentTemp(airCondition.getCurrentTemperature());
        s1.setEndTime(null);
        //自动设置模式

s1.setMode(getMode(airCondition.getTargetTemperature()));
        s1.setStartTime(new Date(systemClock.getTime()));

airCondition.setTargetTemperature(airCondition.getTargetTemperature());
        airCondition.setSpeed(waitObject.getSpeed());
        airCondition.setMode(s1.getMode());

        return;
    }else{
        //选择出所有优先级低于传入的等待对象优先级的服务对象
        if (s1.getSpeed() < waitObject.getSpeed() &&
s1.getSpeed().equals(SPEED_M)){
            serveObjectM.add(s1);
        }
        if (s1.getSpeed() < waitObject.getSpeed() &&
s1.getSpeed().equals(SPEED_L)){
            serveObjectL.add(s1);
        }
    }
}

ServeObject serveObject = null;
//从低风速队列中选择
if (! serveObjectL.isEmpty()){
    serveObject = serveObjectL.get(0);
    //选取服务时长最长的服务对象
    for (ServeObject s2:serveObjectL) {
        if (s2.getServeTime() > serveObject.getServeTime()){
            serveObject = s2;
        }
    }
} else if (! serveObjectM.isEmpty()){
    serveObject = serveObjectM.get(0);
    //选取服务时长最长的服务对象
    for (ServeObject s2:serveObjectM) {
        if (s2.getServeTime() > serveObject.getServeTime()){
            serveObject = s2;
        }
    }
}

//选择出替换的服务对象
if (serveObject != null){
    waitObject waitObjectNew = new waitObject();
    waitObjectNew.setAcId(serveObject.getAcId());
    waitObjectNew.setSpeed(serveObject.getSpeed());
    waitObjectNew.setTimeToWait(TIME_TO_WAIT);

//
waitObjectNew.setTargetTemp(airConditionList.getAirCondition(serveObject.getAcId()).getTargetTemperature());
    serveObject.setEndTime(systemClock);

```

```

        serveMapper.insert(serveObject);
        //写入调度出队列记录
        scheduleLogMapper.insert(new
ScheduleLog(serveObject.getAcId(), SCHEDULE_OUT,
serveObject.getServeId(), systemClock));
        //设置对应的空调状态为待机，未被服务
        airConditionList.setAirConditionOn(waitObjectNew.getAcId());
        //等待队列中不存在该空调的请求，插入等待队列， 若存在则丢弃该请求，旧情
求丢弃
        if (! waitQueue.check(waitObjectNew.getAcId())){
            waitQueue.insert(waitObjectNew);
        }
        //设置新服务对象的状态
        serveObject.setAcId(waitObject.getAcId());

        serveObject.setMode(getMode(airCondition.getTargetTemperature()));
        //设置服务对象
        serveObject.setSpeed(waitObject.getSpeed());
        serveObject.setState(SERVE_RUNNING);

        serveObject.setStartTemp(airCondition.getCurrentTemperature());
        serveObject.setStartTime(new Date(systemClock.getTime()));
        serveObject.setEndTime(null);
        serveObject.setFeeRate(feeMap.get(waitObject.getSpeed()));
        serveObject.setFee(new BigDecimal("0.0"));
        serveObject.setServeTime(0);

        serveObject.setCurrentTemp(airCondition.getCurrentTemperature());

        serveObject.setTargetTemp(airCondition.getTargetTemperature());
        //设置空调
        airCondition.setState(STATE_RUNNING);
        airCondition.setMode(serveObject.getMode());
        airCondition.setSpeed(waitObject.getSpeed());
        //写入调度入队列记录
        scheduleLogMapper.insert(new
ScheduleLog(serveObject.getAcId(), SCHEDULE_IN,
serveObject.getServeId(), systemClock));
    }
}

}

/**
 * 若有空闲直接替换包括等待时长\<=0的等待对象和等待时长\>0的等待对象
 * 若没有空闲,对于等待时长<=0使用时间片调度服务队列中同等级或低等级的服务对象，并给替换下来
的服务对象设置同样的等待时长
 * 没有空闲、替换结束的标志：无更多满足的等待时长<=0的等待对象或 无法替换任何服务时长>0的
服务对象
 */
private void schedule(){
    List<waitObject> failChange = new ArrayList<>();
    waitObject waitObject = waitQueue.schedule();
    while (waitObject != null){
        Integer serveId = serveQueue.hasIdle();
        //存在空闲服务对象
        if (serveId != -1){
            join(serveId, waitObject);

```

```

//          waitQueue.delete(waitObject);
    }else{
        //无空闲服务对象

        //优先级比服务队列中的某些服务对象优先级高
        if(serveQueue.comparePriority(waitObject.getSpeed()) > 0){
            System.out.println("替换高等级");
            AirCondition airCondition =
airConditionList.getAirCondition(waitObject.getAcId());
            //保存中风速的服务对象
            List<ServeObject> serveObjectM = new ArrayList<>();
            //保存低风速的服务对象
            List<ServeObject> serveObjectL = new ArrayList<>();
            for (ServeObject s1: serveQueue.getServeObjectList()) {
                //如果旧请求在被服务，新请求被调度到服务队列
                if (s1.getAcId().equals(waitObject.getAcId())) {
                    //旧请求提前结束，写入数据库
                    s1.setEndTime(systemClock);
                    serveMapper.insert(s1);
                    //新请求替换旧请求所在服务对象
                    s1.setServeTime(0);
                    s1.setFee(new BigDecimal("0.0"));
                    s1.setFeeRate(feeMap.get(waitObject.getSpeed()));

                    s1.setTargetTemp(airCondition.getTargetTemperature());
                    s1.setSpeed(waitObject.getSpeed());

                    s1.setStartTemp(airCondition.getCurrentTemperature());

                    s1.setCurrentTemp(airCondition.getCurrentTemperature());
                    s1.setEndTime(null);
                    //自动设置模式

                    s1.setMode(getMode(airCondition.getTargetTemperature()));
                    s1.setStartTime(new Date(systemClock.getTime()));
//          airCondition.setTargetTemperature();
                    airCondition.setSpeed(waitObject.getSpeed());
                    airCondition.setMode(s1.getMode());
                    return;
                }else{
                    //选择出所有优先级低于传入的等待对象优先级的服务对象
                    if (s1.getSpeed() < waitObject.getSpeed() &&
s1.getSpeed().equals(SPEED_M)){
                        serveObjectM.add(s1);
                    }
                    if (s1.getSpeed() < waitObject.getSpeed() &&
s1.getSpeed().equals(SPEED_L)){
                        serveObjectL.add(s1);
                    }
                }
            }
            ServeObject serveObject = null;
            //从低风速队列中选择
            if (! serveObjectL.isEmpty()){
                serveObject = serveObjectL.get(0);
                //选取服务时长最长的服务对象
                for (ServeObject s2:serveObjectL) {
                    if (s2.getServeTime() > serveObject.getServeTime()){

```

```

        serveObject = s2;
    }
}
} else if (! serveObjectM.isEmpty()){
    serveObject = serveObjectM.get(0);
    //选取服务时长最长的服务对象
    for (ServeObject s2:serveObjectM) {
        if (s2.getServeTime() > serveObject.getServeTime()){
            serveObject = s2;
        }
    }
}
//选择出替换的服务对象
if (serveObject != null){
    waitObject waitObjectNew = new waitObject();
    waitObjectNew.setAcId(serveObject.getAcId());
    waitObjectNew.setSpeed(serveObject.getSpeed());
    waitObjectNew.setTimeToWait(TIME_TO_WAIT);

    //
    waitObjectNew.setTargetTemp(airConditionList.getAirCondition(serveObject.getAcId()).getTargetTemperature());
    serveObject.setEndTime(systemClock);
    serveMapper.insert(serveObject);
    //写入调度出队列记录
    scheduleLogMapper.insert(new
ScheduleLog(serveObject.getAcId(), SCHEDULE_OUT,
serveObject.getServeId(), systemClock));
    //设置对应的空调状态为开机，未被服务

    airConditionList.setAirConditionOn(waitObjectNew.getAcId());
    //等待队列中不存在该空调的请求，插入等待队列， 若存在则丢弃该请求，
旧请求丢弃

    if (! waitQueue.check(waitObjectNew.getAcId())){
        waitQueue.insert(waitObjectNew);
    }
    //设置新服务对象的状态
    serveObject.setAcId(waitObject.getAcId());

    serveObject.setMode(getMode(airCondition.getTargetTemperature()));
    //设置服务对象
    serveObject.setSpeed(waitObject.getSpeed());
    serveObject.setState(SERVE_RUNNING);

    serveObject.setStartTemp(airCondition.getCurrentTemperature());
    serveObject.setStartTime(new
Date(systemClock.getTime()));
    serveObject.setEndTime(null);

    serveObject.setFeeRate(feeMap.get(waitObject.getSpeed()));
    serveObject.setFee(new BigDecimal("0.0"));
    serveObject.setServeTime(0);

    serveObject.setCurrentTemp(airCondition.getCurrentTemperature());

    serveObject.setTargetTemp(airCondition.getTargetTemperature());
    //设置空调
    airCondition.setState(STATE_RUNNING);

```



```

//
airCondition.setScheduleTime(airCondition.getScheduleTime()+1);
                airCondition.setMode(serveObject.getMode());
                airCondition.setSpeed(waitObject.getSpeed());

//
airCondition.setTargetTemperature(waitObject.getTargetTempt());
                //写入调度入队列记录
                scheduleLogMapper.insert(new
ScheduleLog(serveObject.getAcId(),SCHEDULE_IN,
serveObject.getServeId(),systemClock));
//
                waitQueue.delete(waitObject);
        }
        }else if (serveQueue.comparePriority(waitObject.getSpeed()) == 0
&& waitObject.getTimeToWait() <= 0) {
            System.out.println("替换同等级");
            //优先级比服务队列中的某些服务对象优先级相等
            ServeObject serveObject = null;
            for (ServeObject s : serveQueue.getServeObjectList()) {
                if (s.getSpeed().equals(waitObject.getSpeed()) &&
s.getServeTime() != 0) {
                    serveObject = s;
                }
            }
            if (serveObject != null) {
                for (ServeObject s : serveQueue.getServeObjectList()) {
                    if (serveObject.getSpeed().equals(s.getSpeed()) &&
serveObject.getServeTime() < s.getServeTime()) {
                        serveObject = s;
                    }
                }
                //替换
                AirCondition airCondition =
airConditionList.getAirCondition(waitObject.getAcId());
                waitObject waitObjectNew = new waitObject();
                waitObjectNew.setAcId(serveObject.getAcId());
                waitObjectNew.setSpeed(serveObject.getSpeed());
                waitObjectNew.setTimeToWait(TIME_TO_WAIT);

//
waitObjectNew.setTargetTempt(airConditionList.getAirCondition(serveObject.getAcI
d()).getTargetTemperature());
                serveObject.setEndTime(systemClock);
                serveMapper.insert(serveObject);
                //写入调度出队列记录
                scheduleLogMapper.insert(new
ScheduleLog(serveObject.getAcId(), SCHEDULE_OUT, serveObject.getServeId(),
systemClock));

                //设置对应的空调状态为开机，未被服务

                airConditionList.setAirConditionOn(waitObjectNew.getAcId());
                //等待队列中不存在该空调的请求，插入等待队列， 若存在则丢弃该请求，
旧请求丢弃

                if (!waitQueue.check(waitObjectNew.getAcId())) {
                    waitQueue.insert(waitObjectNew);
                }
                //设置新服务对象的状态
                serveObject.setAcId(waitObject.getAcId());

                serveObject.setMode(getMode(airCondition.getTargetTemperature()));

```

```

        //设置服务对象
        serveObject.setSpeed(waitObject.getSpeed());
        serveObject.setState(SERVE_RUNNING);

        serveObject.setStartTemp(airCondition.getCurrentTemperature());
        serveObject.setStartTime(new
Date(systemClock.getTime()));
        serveObject.setEndTime(null);

        serveObject.setFeeRate(feeMap.get(waitObject.getSpeed()));
        serveObject.setFee(new BigDecimal("0.0"));
        serveObject.setServeTime(0);

        serveObject.setCurrentTemp(airCondition.getCurrentTemperature());

        serveObject.setTargetTemp(airCondition.getTargetTemperature());
        //设置空调
        airCondition.setState(STATE_RUNNING);
//
airCondition.setScheduleTime(airCondition.getScheduleTime()+1);
        airCondition.setMode(serveObject.getMode());
        airCondition.setSpeed(waitObject.getSpeed());
//
airCondition.setTargetTemperature(waitObject.getTargetTemp());
        //写入调度入队列记录
        scheduleLogMapper.insert(new
ScheduleLog(serveObject.getAcId(), SCHEDULE_IN, serveObject.getServeId(),
systemClock));
    }
    }else{
        //将未替换成功的等待对象置入一个列表，等待插回
        failChange.add(waitObject);
    }
}
    System.out.println("等待队列调度");
    waitObject = waitQueue.schedule();
}
    waitQueue.insert(failChange);
}
}

```

## SwaggerConfig.java

```

package cn.edu.bupt.aircondition.config;

import com.github.xiaoymin.knife4j.spring.annotations.EnableKnife4j;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import springfox.documentation.builders.ApiInfoBuilder;
import springfox.documentation.builders.PathSelectors;
import springfox.documentation.builders.RequestHandlerSelectors;
import springfox.documentation.spi.DocumentationType;
import springfox.documentation.spring.web.plugins.Docket;

```

```

import springfox.documentation.swagger2.annotations.EnableSwagger2;

@Configuration
@EnableSwagger2
@Enableknife4j
public class SwaggerConfig{
    @Bean
    public Docket createDemoApi() {
        return new Docket(DocumentationType.SWAGGER_2)
            .apiInfo(new ApiInfoBuilder()
                .title("分布式空调系统")
                .description("接口文档")
                .version("1.0")
                .build())//设置API基本信息
            .select()// 选择那些路径和api会生成document

        .apis(RequestHandlerSelectors.basePackage("cn.edu.bupt.aircondition")) // 设置对哪
        些包内的api进行监控
            .paths(PathSelectors.any()) // 对所有路径进行监控
            .build()
            .groupName("v1");
    }
}

```

## WebMvcConfig.java

```

package cn.edu.bupt.aircondition.config;

import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.CorsRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;

@Configuration
public class WebMvcConfig extends WebMvcConfigurerAdapter {
    @Override
    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping("/**")
            .allowedOrigins("**")
            .allowedMethods("POST", "GET", "PUT", "OPTIONS", "DELETE")
            .maxAge(3600)
            .allowCredentials(true);
    }
}

```

## 实体层:

### AirCondition.java

```

package cn.edu.bupt.aircondition.entity;

import com.baomidou.mybatisplus.annotation.TableId;
import lombok.Data;

```

```

import java.math.BigDecimal;

@Data
public class AirCondition {

    @TableId("acId")
    private Integer acId;
    /**
     * @value: 1 开启, 0 关闭, 2 运行中
     */
    private Integer state;
    /**
     * 自动设置
     * @value: 0 制冷, 1制热
     */
    private Integer mode;
    /**
     * @value: 0 低速, 1 中速, 2高速
     */
    private Integer speed;

    private BigDecimal currentTemperature;

    private BigDecimal targetTemperature;

    private BigDecimal originTemperature;

    //总费用: 空调从开机到关机总费用
    private BigDecimal fee;

}

```

## AirConditionList.java

```

package cn.edu.bupt.aircondition.entity;

import lombok.Data;

import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.List;

import static cn.edu.bupt.aircondition.entity.Constant.*;

@Data
public class AirConditionList {
    private List<AirCondition> airConditionList;

    public AirConditionList(){
        airConditionList = new ArrayList<>(NUM_OF_AIR_CONDITION);
        for (int i = 0; i < NUM_OF_AIR_CONDITION; i++) {
            AirCondition airCondition = new AirCondition();
            airCondition.setAcId(i);
            airCondition.setState(STATE_OFF);
            airConditionList.add(i, airCondition);
        }
    }
}

```

```

    }
}

/**
 * 设置默认开机参数
 */
public void setDefaultPara(BigDecimal temptDefault, Integer modeDefault,
    BigDecimal targetTempt, Integer speedDefault, Integer state, BigDecimal fee){
    //设置空调默认风速， 默认初始温度， 默认当前温度， 默认目标温度， 默认模式
    for (AirCondition a:airConditionList) {
        a.setCurrentTemperature(temptDefault);
        a.setOriginTemperature(temptDefault);
        a.setMode(modeDefault);
        a.setTargetTemperature(targetTempt);
        a.setSpeed(speedDefault);
        a.setState(state);
        a.setFee(fee);
    }
}

/**
 * 关机时的处理
 */
public void setOff(){
    for (AirCondition a:airConditionList) {
        a.setState(STATE_OFF);
    }
}

/**
 * 获得房间状态
 */
public AirCondition getAirCondition(Integer acId){
    return airConditionList.get(acId);
}

/**
 * 设置空调的状态为On
 */
public void setAirConditionOn(Integer acId){
    AirCondition airCondition = airConditionList.get(acId);
    airCondition.setState(STATE_ON);
}
}

```

## AirConditionLog.java

```

package cn.edu.bupt.aircondition.entity;

import lombok.Data;

import java.util.Date;

/**
 * 空调开关机请求 存储数据库的实体
 */
@Data
public class AirConditionLog {

```

```

private Integer acId;    //空调ID

private Integer state;   //采用Constant中的常量STATE_ON, STATE_OFF

private Date date;      //状态变化时的系统时间

public AirConditionLog(Integer acId, Integer state, Date date){
    this.acId = acId;
    this.date = date;
    this.state = state;
}
}

```

### BillsWithDetails.java

```

package cn.edu.bupt.aircondition.entity;

import lombok.Data;

import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

@Data
public class BillswithDetails {
    /**
     * 账单及详单类
     * create by zhicheng lee
     * */
    private Integer customerId;
    private String name;
    private Integer acId;//room id OR air condition id
    private Date startTime;
    private Date endTime;
    private BigDecimal allConsumption;
    private BigDecimal allFee;

    private List<DetailObject> detailObjectList;

    public BillswithDetails() {
        this.detailObjectList=new ArrayList<>();
    }
}

```

### Constant.java

```

package cn.edu.bupt.aircondition.entity;

import java.math.BigDecimal;
import java.util.Date;
import java.util.HashMap;

```

```

import java.util.Map;

/**
 * @about 系统相关的常量,
 */
public class Constant {
    public final static Integer SPEED_H = 2; //高风速
    public final static Integer SPEED_M = 1; //中风速
    public final static Integer SPEED_L = 0; //低风速

    public final static Integer STATE_ON = 1; //空调开启
    public final static Integer STATE_OFF = 2; //空调关闭
    public final static Integer STATE_RUNNING = 3; //空调运行

    public final static Integer MODE_H = 1; //制热
    public final static Integer MODE_R = 0; //制冷

    public final static Integer SERVE_IDLE = 0; //服务对象空闲
    public final static Integer SERVE_RUNNING = 1; // 服务对象服务中

    public final static Integer NUM_OF_AIR_CONDITION= 5;

    public static Integer SPEED_DEFAULT; //空调默认风速

    public static BigDecimal TEMPT_DEFAULT = new BigDecimal("22.0"); //空调默认温度

    public static Integer MODE_DEFAULT; //空调默认模式

    public static BigDecimal TEMPT_HIGH_LIMIT; //空调默认最高温度

    public static BigDecimal TEMPT_LOW_LIMIT; //空调默认最低温度

    public static Map<Integer,BigDecimal> feeMap = new HashMap<>(); //费用
    public static Map<Integer,BigDecimal> temptMap = new HashMap<>(); //温度

    //调度
    public final static Integer SCHEDULE_IN = 1; //调入服务队列
    public final static Integer SCHEDULE_OUT = 2; //调出服务队列

    //默认分配等待时长2分钟
    public final static Integer TIME_TO_WAIT = 2;

    //改为指定系统开始时间
    public static Date systemClock = new Date(System.currentTimeMillis());

    //报表类型
    public final static Integer REPORT_DAY = 1; //日报表
    public final static Integer REPORT_WEEK = 2; //周报表
    public final static Integer REPORT_MONTH = 3; //月报表
    public final static Integer REPORT_YEAR = 4; //年报表

    //是否到达目标温度
    public final static Integer REACH_TARGET = 1;
    public final static Integer NO_REACH_TARGET = 0;

    /**

```

```

        *
        * @param tempt 请求的温度
        * @return 返回对应的模式
        */
    public static Integer getMode(BigDecimal tempt){
        return MODE_DEFAULT;
    }
}

```

## DetailObject.java

```

package cn.edu.bupt.aircondition.entity;

import lombok.Data;

import java.math.BigDecimal;
import java.util.Date;

@Data
public class DetailObject {
    /**
     * 详单类
     */

    private Integer acId;
    private Integer speed; //value: 0 低速, 1 中速, 2高速
    private Integer mode; //value: 0 制冷, 1制热
    private Date startTime;
    private Date endTime;
    private BigDecimal startTempt;
    private BigDecimal endTempt;
    private BigDecimal consumption; //高风: 1度/1分钟 中风: 1度/2分钟 低风: 1度/3分钟
    private BigDecimal fee;

    public DetailObject() {}
}

```

## ScheduleLog.java

```

package cn.edu.bupt.aircondition.entity;

import lombok.Data;

import java.util.Date;

@Data
public class ScheduleLog {

    private Integer acId; //空调ID

    private Integer type; //被调度类型: 1 调出, 2 调入
}

```



```

        private Integer serveId;    // 若为调出，表示之前服务的服务对象， 若为调入， 表示将要
        服务的服务对象

        private Date date;    // 被调度时，系统时间

        public ScheduleLog(Integer acId, Integer type, Integer serveId, Date date){
            this.acId = acId;
            this.date = date;
            this.serveId = serveId;
            this.type = type;
        }
    }
}

```

## ServeQueue.java

```

package cn.edu.bupt.aircondition.entity;

import lombok.Data;

import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.List;

import static cn.edu.bupt.aircondition.entity.Constant.*;

/**
 * @about: 服务队列
 */
@Data
public class ServeQueue {

    private List<ServeObject> serveObjectList;

    /**
     * 初始化
     * @param numOfServeObject
     */
    public void init(Integer numOfServeObject){
        serveObjectList = new ArrayList<>();
        for (int i = 0; i < numOfServeObject ; i++) {
            ServeObject serveObject = new ServeObject();
            serveObject.setServeId(i);
            serveObjectList.add(i, serveObject);
        }
    }

    /**
     * 设置默认模式
     * @param mode 模式
     * @param speed 风速
     * @param feeRate 费率
     * @param startTempt 开始
     * @param currentTempt
     * @param fee
     */
}

```

```

    public void setDefaultPara(Integer mode, Integer speed, BigDecimal feeRate,
        BigDecimal startTemp, BigDecimal currentTemp, BigDecimal targetTemp,
        BigDecimal fee){
        for (ServeObject s:serveObjectList) {
            s.setTargetTemp(targetTemp);
            s.setMode(mode);
            s.setSpeed(speed);
            s.setFeeRate(feeRate);
            s.setStartTemp(startTemp);
            s.setCurrentTemp(currentTemp);
            s.setFee(fee);
        }
    }

    /**
     * 用于初始化时，设置所有服务对象空闲
     */
    public void setReady(){
        for (ServeObject s:serveObjectList) {
            s.setState(Constant.SERVE_IDLE);
        }
    }

    /**
     *
     *
     * @return -1 表示没有空闲的服务对象，[0-numOfServeObject-1]表示有空闲的服务对象，返
     回值表示服务对象ID，也是服务对象在服务队列中的位置
     */
    public Integer hasIdle(){
        for (ServeObject s:serveObjectList) {
            if (s.getState().equals(SERVE_IDLE)){
                return s.getServeId();
            }
        }
        return -1;
    }

    /**
     *
     * @param speed 请求服务的风速
     * @return 1 大于某个/某些服务对象优先级 0 等于某个/某些服务对象优先级 -1小于所有服务对
     象优先级
     */
    public Integer comparePriority(Integer speed){
        Integer cSpeed = SPEED_H;
        for (ServeObject s:serveObjectList) {
            if (s.getSpeed() < cSpeed){
                cSpeed = s.getSpeed();
            }
        }
        if (cSpeed > speed){
            return -1;
        }else if (cSpeed.equals(speed)){
            return 0;
        }else{
            return 1;
        }
    }
}

```

```

/**
 * 获取服务对象
 */
public ServeObject getServeObject(Integer serveId){
    return serveObjectList.get(serveId);
}
/**
 * 检查是否空调被服务
 * @return null 没有被服务， 正在提供服务的服务对象
 */
public ServeObject check(Integer acId){
    for (ServeObject s:serveObjectList) {
        if (s.getState().equals(SERVE_RUNNING) &&s.getAcId().equals(acId)){
            return s;
        }
    }
    return null;
}
}

```

## ServeMapper.java

```

package cn.edu.bupt.aircondition.mapper;

import cn.edu.bupt.aircondition.entity.ServeObject;
import com.baomidou.mybatisplus.core.mapper.BaseMapper;
import org.apache.ibatis.annotations.Param;
import org.apache.ibatis.annotations.Select;
import org.springframework.stereotype.Repository;

import java.util.Date;
import java.util.List;

@Repository
public interface ServeMapper extends BaseMapper<ServeObject> {

    //获取最常用目标温度
    @Select("SELECT A.target_temp FROM (SELECT target_temp, SUM(serve_time) as max_serve_time FROM serve_object WHERE ac_id = #{acId} AND start_time >= #{startTime} AND end_time <= #{endTime} GROUP BY target_temp ORDER BY max_serve_time DESC LIMIT 1) as A")
    Double getCommonTargetTemp(@Param("acId") Integer acId, @Param("startTime")Date startTime, @Param("endTime")Date endTime);

    //获取最常用风速
    @Select("SELECT A.speed FROM (SELECT speed, SUM(serve_time) as max_serve_time FROM serve_object WHERE ac_id = #{acId} AND start_time >= #{startTime} AND end_time <= #{endTime} GROUP BY speed ORDER BY max_serve_time DESC LIMIT 1) as A")
    Integer getCommonSpeed(@Param("acId") Integer acId, @Param("startTime") Date startTime, @Param("endTime") Date endTime);

    //达到目标温度的次数
    @Select("SELECT COUNT(*) FROM serve_object WHERE ac_id = #{acId} AND start_time >= #{startTime} AND end_time <= #{endTime} AND current_temp = target_temp")
}

```

```

    Integer getReachTargetTimes(@Param("acId") Integer acId, @Param("startTime")
Date startTime, @Param("endTime") Date endTime);
    //详单数
    @Select("SELECT COUNT(*) FROM serve_object WHERE ac_id = #{acId} AND
start_time >= #{startTime} AND end_time <= #{endTime}")
    Integer getDetailCount(@Param("acId") Integer acId, @Param("startTime") Date
startTime, @Param("endTime") Date endTime);
    //花费
    @Select("SELECT SUM(fee) FROM serve_object WHERE ac_id = #{acId} AND
start_time >= #{startTime} AND end_time <= #{endTime}")
    Double getFee(@Param("acId") Integer acId, @Param("startTime") Date
startTime, @Param("endTime") Date endTime);

    @Select("SELECT * FROM serve_object WHERE ac_id = #{acId} AND start_time >=
#{startTime} AND end_time <= #{endTime}")
    List<ServeObject> getDetail(@Param("acId") Integer acId, @Param("startTime")
Date startTime, @Param("endTime") Date endTime);
}

```

## 前端代码：

### admin.html

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>空调管理员</title>
    <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/jquery@3.2/dist/jquery.min.js">
</script>
    <script src="https://cdn.staticfile.org/jquery-
cookie/1.4.1/jquery.cookie.min.js"></script>
    <script>
        var timer;
        var stateToText = ["关闭", "待机", "关闭", "运行"];
        var speedToText = ["低风速", "中风速", "高风速"];
        var modeToText = ["制冷", "制热"];
        var serveStateToText = ["空闲", "服务中"];

        function getState() {
            $.ajax({
                url: "http://119.3.212.133:20002/admin/state",
                type: "GET",
                success: function (result) {
                    var list = result.result.airConditionList.airConditionList;
                    var serve = result.result.serveQueue.serveObjectList;
                    var L = result.result.waitQueue.queueL;
                    var M = result.result.waitQueue.queueM;
                    var H = result.result.waitQueue.queueH;
                    var i = 1;

```

```

        for (i = 1; i <= 5; i++) {
            var ac = list[i - 1];

            document.getElementById("acList").rows[i].cells[1].innerHTML =
stateToText[ac.state];

            document.getElementById("acList").rows[i].cells[2].innerHTML =
modeToText[ac.mode];

            document.getElementById("acList").rows[i].cells[3].innerHTML =
speedToText[ac.speed];

            document.getElementById("acList").rows[i].cells[4].innerHTML =
ac.currentTemperature;

            document.getElementById("acList").rows[i].cells[5].innerHTML =
ac.targetTemperature;

            document.getElementById("acList").rows[i].cells[6].innerHTML =
ac.originTemperature;

            document.getElementById("acList").rows[i].cells[7].innerHTML = ac.fee;
        }
        for (i = 1; i <= serve.length; i++) {
            var ac = serve[i - 1];

            document.getElementById("serveQ").rows[i].cells[0].innerHTML = ac.acId;

            document.getElementById("serveQ").rows[i].cells[1].innerHTML = ac.serveId;

            document.getElementById("serveQ").rows[i].cells[2].innerHTML =
serveStateToText[ac.state];

            document.getElementById("serveQ").rows[i].cells[3].innerHTML =
modeToText[ac.mode];

            document.getElementById("serveQ").rows[i].cells[4].innerHTML =
speedToText[ac.speed];

            document.getElementById("serveQ").rows[i].cells[5].innerHTML = ac.startTemp;

            document.getElementById("serveQ").rows[i].cells[6].innerHTML = ac.targetTemp;

            document.getElementById("serveQ").rows[i].cells[7].innerHTML = ac.currentTemp;

            document.getElementById("serveQ").rows[i].cells[8].innerHTML = ac.feeRate;

            document.getElementById("serveQ").rows[i].cells[9].innerHTML = ac.fee;

            document.getElementById("serveQ").rows[i].cells[10].innerHTML = ac.startTime;

            document.getElementById("serveQ").rows[i].cells[11].innerHTML = ac.endTime;

            document.getElementById("serveQ").rows[i].cells[12].innerHTML = ac.serveTime;
        }
        for (i = 1; H[i - 1] != null; i++) {
            var ac = H[i - 1];

```

```

        document.getElementById("H").rows[i].cells[0].innerHTML
= ac.acId;
        document.getElementById("H").rows[i].cells[1].innerHTML
= ac.speed;
        document.getElementById("H").rows[i].cells[2].innerHTML
= ac.timeToWait;
    }
    for (i = 1; M[i - 1] != null; i++) {
        var ac = M[i - 1];
        document.getElementById("M").rows[i].cells[0].innerHTML
= ac.acId;
        document.getElementById("M").rows[i].cells[1].innerHTML
= ac.speed;
        document.getElementById("M").rows[i].cells[2].innerHTML
= ac.timeToWait;
    }
    for (i = 1; L[i - 1] != null; i++) {
        var ac = L[i - 1];
        document.getElementById("L").rows[i].cells[0].innerHTML
= ac.acId;
        document.getElementById("L").rows[i].cells[1].innerHTML
= ac.speed;
        document.getElementById("L").rows[i].cells[2].innerHTML
= ac.timeToWait;
    }

    },
    error: function (jqXHR, textStatus, errorThrown) {
        alert("错误");
        clearInterval(timer);
    }
});

}
$(document).ready(function () {
    $("#defaultMode").bind("input propertychange", function () {
        if ($("#defaultMode").val() == 1) {
            $("#defaultTemp").val(28);
            $("#temptHighLimit").val(30);
            $("#temptLowLimit").val(25);
        }
        else if ($("#defaultMode").val() == 0) {
            $("#defaultTemp").val(22);
            $("#temptLowLimit").val(18);
            $("#temptHighLimit").val(25);
        }
    });

    $("#start-btn").click(function () {
        $.ajax({
            url: "http://119.3.212.133:20002/admin/start",
            type: "GET",
            data: {
                numOfServeObject: $("#numOfServeObject").val(),

```

```

        defaultMode: $("#defaultMode").val(),
        defaultSpeed: $("#defaultSpeed").val(),
        defaultTemp: $("#defaultTemp").val(),
        feeRateH: $("#feeRateH").val(),
        feeRateL: $("#feeRateL").val(),
        feeRateM: $("#feeRateM").val(),
        temptHighLimit: $("#temptHighLimit").val(),
        temptLowLimit: $("#temptLowLimit").val()
    },
    success: function (result) {
        if (result.code == 200) {
            alert("成功开启");
        } else {
            alert("中央空调已开机");
        }
    },
    error: function (jqXHR, textStatus, errorThrown) {
        alert("错误");
    }
});
});
$("#state-btn").click(function(){
    console.log($("#frequent").val());
    var t = $("#frequent").val()*1000;
    clearInterval(timer);
    timer = setInterval(getState, t);
});
$("#off-btn").click(function () {
    $.ajax({
        url: "http://119.3.212.133:20002/admin/off",
        type: "GET",
        success: function (result) {
            alert("空调已关闭");
            clearInterval(timer);
        },
        error: function (jqXHR, textStatus, errorThrown) {
            alert("错误");
        }
    });
});
})
</script>
</head>

<body>
    <div class="container">
        <div class="row clearfix">
            <div class="col-md-8 column">
                <form class="form-horizontal" role="form">
                    <div class="form-group">
                        <label for="numOfServeObject" class="col-sm-2 control-label">服务队列数</label>
                        <div class="col-sm-10">
                            <input name="numOfServeObject" type="email"
                                class="form-control" id="numOfServeObject"
                                value="3" />
                        </div>
                    </div>
                </form>
            </div>

```

```

        <div class="form-group">
            <label for="defaultMode" class="col-sm-2 control-label">
默认模式</label>

            <div class="col-sm-10">
                <select class="form-control" id="defaultMode">
                    <option title="制冷" selected value="0">制冷
</option>

                    <option title="制热" value="1">制热</option>
                </select>
            </div>
        </div>
        <div class="form-group">
            <label for="defaultSpeed" class="col-sm-2 control-label">默认风速</label>

            <div class="col-sm-10">
                <select class="form-control" id="defaultSpeed">
                    <option title="低风" value="0">低风</option>
                    <option title="中风" value="1" selected>中风
</option>

                    <option title="高风" value="2">高风</option>
                </select>
            </div>
        </div>
        <div class="form-group">
            <label for="defaultTemp" class="col-sm-2 control-label">缺省温度</label>

            <div class="col-sm-10">
                <input name="defaultTemp" class="form-control"
id="defaultTemp" value="22" />
            </div>
        </div>
        <div class="form-group">
            <label for="feeRateL" class="col-sm-2 control-label">低速
费用率</label>

            <div class="col-sm-10">
                <input name="feeRateH" class="form-control"
id="feeRateL" value="4" />
            </div>
        </div>
        <div class="form-group">
            <label for="feeRateM" class="col-sm-2 control-label">中速
费用率</label>

            <div class="col-sm-10">
                <input name="feeRateL" class="form-control"
id="feeRateM" value="5" />
            </div>
        </div>
        <div class="form-group">
            <label for="feeRateH" class="col-sm-2 control-label">高速
费用率</label>

            <div class="col-sm-10">
                <input name="feeRateM" class="form-control"
id="feeRateH" value="6" />
            </div>
        </div>
        <div class="form-group">
            <label for="tempHighLimit" class="col-sm-2 control-label">最高温度</label>

```



```

        <div class="col-sm-10">
            <input name="temptHighLimit" class="form-control"
id="temptHighLimit" value="25" />
        </div>
    </div>
    <div class="form-group">
        <label for="temptLowLimit" class="col-sm-2 control-label">最低温度</label>
        <div class="col-sm-10">
            <input name="temptLowLimit" class="form-control"
id="temptLowLimit" value="18" />
        </div>
    </div>
</form>
</div>
<div class="col-md-4 column">
    <button id="start-btn" type="button" class="btn btn-default btn-block btn-success">开启中央空调</button>
    <button id="off-btn" type="button" class="btn btn-block btn-warning">关闭中央空调</button>
</div>
</div>

<div class="row clearfix">
    <div class="col-md-8 column">
        <form class="form-horizontal" role="form">
            <div class="form-group">
                <label for="frequent" class="col-sm-2 control-label">刷新
频率</label>

                <div class="col-sm-10">
                    <select class="form-control" id="frequent">
                        <option title="1秒刷新一次">1</option>
                        <option title="2秒刷新一次" selected>2</option>
                        <option title="5秒刷新一次">5</option>
                        <option title="10秒刷新一次">10</option>
                    </select>
                </div>
            </div>
            <div class="col-md-4 column">
                <button id="state-btn" type="button" class="btn btn-default btn-block btn-primary">查看空调状态</button>
            </div>
        </div>

    <div class="row clearfix">
        <div class="col-md-12 column">
            <p>
                <strong>从控机列表</strong>
            </p>
            <table id="acList" class="table">
                <thead>
                    <tr>
                        <th>acId</th>
                        <th>state</th>
                        <th>mode</th>
                        <th>speed</th>

```

[illegible]

```
        </table>
    </div>
</div>
<div class="row clearfix">
    <div class="col-md-12 column">
        <p>
            <strong>服务队列</strong>
        </p>
        <table id="serveq" class="table">
            <thead>
                <tr>
                    <th>acId</th>
                    <th>serveId</th>
                    <th>state</th>
                    <th>mode</th>
                    <th>speed</th>
                    <th>startTemp</th>
                    <th>targetTemp</th>
                    <th>currentTemp</th>
                    <th>feeRate</th>
                    <th>fee</th>
                    <th>startTime</th>
                    <th>endTime</th>
                    <th>serveTime</th>
                </tr>
            </thead>
            <tbody>
                <tr class="warning">
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                </tr>
                <tr class="success">
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                </tr>
                <tr class="info">
```

```

        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
    </tr>
</tbody>
</table>
</div>
</div>
<p><strong>等待队列</strong></p>
<div class="row clearfix">
    <div class="col-md-4 column">
        <p><strong>高风速等待队列</strong></p>
        <table id="H" class="table">
            <thead>
                <tr>
                    <th>acId</th>
                    <th>speed</th>
                    <th>timeTowait</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td></td>
                    <td></td>
                    <td></td>
                </tr>
                <tr class="success">
                    <td></td>
                    <td></td>
                    <td></td>
                </tr>
                <tr class="error">
                    <td></td>
                    <td></td>
                    <td></td>
                </tr>
                <tr class="warning">
                    <td></td>
                    <td></td>
                    <td></td>
                </tr>
                <tr class="info">
                    <td></td>
                    <td></td>
                    <td></td>
                </tr>
            </tbody>
        </table>
    </div>
</div>

```

```
</div>
<div class="col-md-4 column">
  <p><strong>中风速等待队列</strong></p>
  <table id="M" class="table">
    <thead>
      <tr>
        <th>acId</th>
        <th>speed</th>
        <th>timeToWait</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td></td>
        <td></td>
        <td></td>
      </tr>
      <tr class="success">
        <td></td>
        <td></td>
        <td></td>
      </tr>
      <tr class="error">
        <td></td>
        <td></td>
        <td></td>
      </tr>
      <tr class="warning">
        <td></td>
        <td></td>
        <td></td>
      </tr>
      <tr class="info">
        <td></td>
        <td></td>
        <td></td>
      </tr>
    </tbody>
  </table>
</div>
<div class="col-md-4 column">
  <p><strong>低风速等待队列</strong></p>
  <table id="L" class="table">
    <thead>
      <tr>
        <th>acId</th>
        <th>speed</th>
        <th>timeToWait</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td></td>
        <td></td>
        <td></td>
      </tr>
      <tr class="success">
        <td></td>
```

```

                <td></td>
                <td></td>
            </tr>
            <tr class="error">
                <td></td>
                <td></td>
                <td></td>
            </tr>
            <tr class="warning">
                <td></td>
                <td></td>
                <td></td>
            </tr>
            <tr class="info">
                <td></td>
                <td></td>
                <td></td>
            </tr>
        </tbody>
    </table>
</div>
</div>
</body>

</html>

```

## adddetailList.html

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>详细账单</title>
    <link rel="stylesheet"
href="https://apps.bding.com/libs/bootstrap/3.2.0/css/bootstrap.min.css">
    <script src="https://cdn.jsdelivr.net/npm/jquery@3.2/dist/jquery.min.js">
</script>
    <script src="https://cdn.staticfile.org/jquery-
cookie/1.4.1/jquery.cookie.min.js"></script>
    <style>
        body {
            background-repeat: no-repeat;
        }
    </style>
</head>

<body onload="load()">
    <div class="container">
        <div class="row clearfix">
            <div class="col-md-4 column">
                <form class="form-horizontal" role="form">
                    <div class="form-group">
                        <label for="date" class="col-sm-2 control-
label">Date</label>

```

```

        <div class="col-sm-10">
            <input class="form-control" id="date" />
        </div>
    </div>
    <div class="form-group">
        <label for="type" class="col-sm-2 control-label">Type</label>
        <div class="col-sm-10">
            <select class="form-control" id="type">
                <option title="日报表" selected value="1">日报表
            </option>
                <option title="周报表" value="2">周报表</option>
                <option title="月报表" value="3">月报表</option>
                <option title="年报表" value="4">年报表</option>
            </select>
        </div>
    </div>
    </form> <button id="repo-btn" type="button" class="btn btn-success btn-block btn-default"
        style="background-color: rgb(34, 108, 245);">查看报表</button>
</div>
<div class="col-md-12 column">
    <div class="panel panel-success">
        <h1 class="panel-title " style="text-align: center;font-size:30px; ">
            详细账单
        </h1>
        <div class="panel-body " style="font-size:15px;background-color: aliceblue; ">
            <table class="table" id="detailList">
                <thead>
                    <tr>
                        <th>
                            服务开始时间
                        </th>
                        <th>
                            服务结束时间
                        </th>
                        <th>
                            风速
                        </th>
                        <th>
                            模式
                        </th>
                        <th>
                            服务开始温度
                        </th>
                        <th>
                            服务结束温度
                        </th>
                        <th>
                            用电量
                        </th>
                        <th>
                            费用
                        </th>
                    </tr>
                </thead>
            </table>
        </div>
    </div>
</div>

```

```

        <tbody>
        </tbody>
    </table>
</div>

</div>

</div>

</div>
</body>

<script type="text/javascript">
    function getQueryVariable(variable) {
        var query = window.location.search.substring(1);
        var vars = query.split("&");
        for (var i = 0; i < vars.length; i++) {
            var pair = vars[i].split("=");
            if (pair[0] == variable) { return pair[1]; }
        }
        return (false);
    }
    acId = getQueryVariable("acId");
    $("#repo-btn").click(function () {
        $.ajax({
            url: "http://119.3.212.133:20002/getDetailList",
            type: "GET",
            data: {
                acId: acId,
                date: $("#date").val(),
                type: $("#type").val()
            },
            success: function (result) {
                var tab = document.getElementById("detailList");
                var row, cell;
                var detailObjectList = result.result.detailObjectList;
                var i;
                var len = detailObjectList.length;
                for (i = 0; i < len; i++) {
                    row = tab.insertRow(); //参数-1表示新增的行或者列是新增在最后面
                    var detailObject = detailObjectList[i];

                    cell = row.insertCell();
                    cell.innerHTML = detailObject.startTime;

                    cell = row.insertCell();
                    cell.innerHTML = detailObject.endTime;

                    cell = row.insertCell();
                    cell.innerHTML = detailObject.speed;

                    cell = row.insertCell();
                    cell.innerHTML = detailObject.mode;
                }
            }
        });
    });

```



```

        cell = row.insertCell();
        cell.innerHTML = detailObject.startTemp;

        cell = row.insertCell();
        cell.innerHTML = detailObject.endTemp;

        cell = row.insertCell();
        cell.innerHTML = detailObject.consumption;

        cell = row.insertCell();
        cell.innerHTML = detailObject.fee;
    }
},
error: function (jqXHR, textStatus, errorThrown) {
    alert("checkout error");
}
});
});

```

</script>

</html>

## detailList.html

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>detailList</title>
    <link rel="stylesheet"
href="https://apps.bdimg.com/libs/bootstrap/3.2.0/css/bootstrap.min.css">
    <script src="https://cdn.jsdelivr.net/npm/jquery@3.2/dist/jquery.min.js">
</script>
    <script src="https://cdn.staticfile.org/jquery-
cookie/1.4.1/jquery.cookie.min.js"></script>
    <style>
        body {
            background-image: url('background.jpg');
            background-repeat: no-repeat;
        }
    </style>
</head>

<body onload="load()">
    <div class="container">
        <div class="row clearfix">
            <div class="col-md-12 column">
                <div class="panel panel-success">

```

```

        <h1 class="panel-title " style="text-align: center;font-size:30px; ">
            请签收您的账单
        </h1>
        <div class="panel-body " style="font-size:15px;background-color: aliceblue; ">
            <table class="table" id="detailList">
                <thead>
                    <tr>
                        <th>
                            服务开始时间
                        </th>
                        <th>
                            服务结束时间
                        </th>
                        <th>
                            风速
                        </th>
                        <th>
                            模式
                        </th>
                        <th>
                            服务开始温度
                        </th>
                        <th>
                            服务结束温度
                        </th>
                        <th>
                            用电量
                        </th>
                        <th>
                            费用
                        </th>
                    </tr>
                </thead>
                <tbody>

                </tbody>
            </table>
        </div>
        <div class="panel-footer" style="font-size:15px;">
            <h3>
                本次需要支付的费用为: <span id="allFee"></span>
            </h3>
            <h3>
                <button type="button" class="btn btn-default btn-block" style="padding: 3px 6px;
                    background: rgb(34, 108, 245);text-align: center;font-size:40px;"
                    onclick="checkout()">确定</button>
            </h3>
        </div>

    </div>

</div>
</div>
</div>

```

```

</div>
</body>

<script type="text/javascript">
    var url = location.search; //获取url中"?"符后的字符串
    if (url.indexOf("?") != -1) { //判断是否有参数
        var str = url.substr(1); //从第一个字符开始 因为第0个是?号 获取所有除问号的所有字符串
        strs = str.split("="); //用等号进行分隔 （因为知道只有一个参数 所以直接用等号进行分隔 如果有多个参数 要用&号分隔 再用等号进行分隔）
        // alert(strs[1]); //直接弹出第一个参数 （如果有多个参数 还要进行循环的）
    }

    acId = strs[1];

    function load() {
        $.ajax({
            url: "http://119.3.212.133:20002/front/checkout",
            type: "GET",
            data: {
                "acId": acId
            },
            success: function(result) {
                // alert("checkout success");
                document.getElementById("allFee").innerHTML = result.result.allFee;
                var tab = document.getElementById("detailList");
                var row, cell;
                var detailObjectList = result.result.detailObjectList;

                var i;
                var len = detailObjectList.length;
                // alert("length:"+len);
                for (i = 0; i < len; i++) {
                    row = tab.insertRow(); //参数-1表示新增的行或者列是新增在最后面
                    var detailObject = detailObjectList[i];

                    // alert("detailObject"+detailObject);

                    cell = row.insertCell();
                    cell.innerHTML = detailObject.startTime;

                    cell = row.insertCell();
                    cell.innerHTML = detailObject.endTime;

                    cell = row.insertCell();
                    cell.innerHTML = detailObject.speed;

                    cell = row.insertCell();
                    cell.innerHTML = detailObject.mode;

                    cell = row.insertCell();

```

```

        cell.innerHTML = detailObject.startTempt;

        cell = row.insertCell();
        cell.innerHTML = detailObject.endTempt;

        cell = row.insertCell();
        cell.innerHTML = detailObject.consumption;

        cell = row.insertCell();
        cell.innerHTML = detailObject.fee;
    }
},
error: function(jqXHR, textStatus, errorThrown) {
    alert("checkout error");
}
})
}

function checkout() {

    window.location.href = './userLogin.html';
}
</script>

</html>

```

## index.html

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <meta charset="utf-8">
    <title>index</title>
    <link href="CSS_styles/main.css" rel="stylesheet">
    <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/jquery@3.2/dist/jquery.min.js">
</script>
    <script src="https://cdn.staticfile.org/jquery-
cookie/1.4.1/jquery.cookie.min.js"></script>
</head>

<body>
    <div id="root">
        <div class="Dashboard fullscreen dark">
            <div class="Background">
                <div class="fullscreen" style="background-color: black; opacity:
1; transition: opacity 200ms ease-in-out 0s;">
                    <div class="Image fullscreen" style="background-image:
url(background.jpg);"></div>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
        <div class="widgets fullscreen">
            <div class="container">
                <div class="Slot middleCentre">
                    <div class="widget " style="font-size: 24px;">
                        <div class="Greeting" style="margin-bottom: 30px;">
                            <h2>欢迎使用空调管理系统</h2>
                        </div>
                    </div>
                    <div class="widget " style="font-size: 24px;">
                        <div class="Links" style="grid-template-columns: 1fr
1fr; text-align: left;">
                            <button type="button" class="btn btn-info"
style="width: 200px;height: 50px;"
                            onclick="window.location.href =
'./userLogin.html'">入住客户</button>
                            <button type="button" class="btn btn-info"
style="width: 200px;height: 50px;"
                            onclick="window.location.href =
'./staffLogin.html'">工作人员</button>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>
</html>

```

## manger.html

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>酒店经理</title>
    <link rel="stylesheet"
href="https://apps.bdimg.com/libs/bootstrap/3.2.0/css/bootstrap.min.css">
    <script src="https://cdn.jsdelivr.net/npm/jquery@3.2/dist/jquery.min.js">
</script>
    <script src="https://cdn.staticfile.org/jquery-
cookie/1.4.1/jquery.cookie.min.js"></script>
    <script>
        function getDeteil(acId){
            var start = document.getElementById("starttime").innerText;
            var end = document.getElementById("endtime").innerText;
            window.location.href="./alldetailList.html?acId="+acId;
        }
        $(document).ready(function() {
            $("#repo-btn").click(function() {
                $.ajax({
                    url: "http://119.3.212.133:20002/manager/report",
                    type: "GET",
                    data: {
                        date: $("#date").val(),

```



酒店经理

```
        </h1>
      </div>
    </div>
  </div>
  <div class="co-md-12 column" style="background-color:
antiquewhite;">
    <div class="row clearfix">
      <div class="col-md-4 column">
        <form class="form-horizontal" role="form">
          <div class="form-group">
            <label for="date" class="col-sm-2 control-
label">Date</label>

            <div class="col-sm-10">
              <input class="form-control" id="date" />
            </div>
          </div>
          <div class="form-group">
            <label for="type" class="col-sm-2 control-
label">Type</label>

            <div class="col-sm-10">
              <select class="form-control" id="type">
                <option title="日报表" selected
value="1">日报表</option>

                <option title="周报表" value="2">周报表
</option>

                <option title="月报表" value="3">月报表
</option>

                <option title="年报表" value="4">年报表
</option>

              </select>
            </div>
          </div>
        </form> <button id="repo-btn" type="button"
class="btn btn-success btn-block btn-default" style="background-color: rgb(34,
108, 245);">查看报表</button>
      </div>
      <div class="col-md-8 column">
        <div class="list-group">
          <div class="list-group-item">
            <h4 class="list-group-item-heading">
              Start-Time
            </h4>
            <p id="starttime" class="list-group-item-
text">

            </p>
          </div>
          <div class="list-group-item">
            <h4 class="list-group-item-heading">
              End-Time
            </h4>
            <p id="endtime" class="list-group-item-
text">

            </p>
          </div>
        </div>
      </div>
    </div>
```

```

    </div>
</div>
<div class="row clearfix">
    <div class="col-md-12 column">
        <p>
            <strong>report</strong>
        </p>
        <table id="repo-table" class="table">
            <thead>
                <tr>
                    <th>
                        acId
                    </th>
                    <th>
                        使用次数
                    </th>
                    <th>
                        最常用目标温度
                    </th>
                    <th>
                        最常用风速
                    </th>
                    <th>到达目标温度次数</th>
                    <th>被调度次数</th>
                    <th>详单记录数</th>
                    <th>总费用</th>
                </tr>
            </thead>
            <tbody>
                <tr onclick="getDeteil(0)">
                    <td>0</td>
                    <td>
                    </td>
                    <td>
                    </td>
                    <td>
                    </td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                </tr>
                <tr class="success" onclick="getDeteil(1)">
                    <td>
                        1
                    </td>
                    <td>
                    </td>
                    <td>
                    </td>
                    <td>
                    </td>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                </tr>
                <tr class="error" onclick="getDeteil(2)">

```



```

        <td>
            2
        </td>
        <td>
        </td>
        <td>
        </td>
        <td>
        </td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
    </tr>
    <tr class="warning" onclick="getDeteil(3)">
        <td>
            3
        </td>
        <td>
        </td>
        <td>
        </td>
        <td>
        </td>
        <td>
        </td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
    </tr>
    <tr class="info" onclick="getDeteil(4)">
        <td>
            4
        </td>
        <td>
        </td>
        <td>
        </td>
        <td>
        </td>
        <td>
        </td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
    </tr>
</tbody>
</table>
</div>
</div>

</div>

<div class="col-md-1 column"></div>

</div>

</div>
```

```

</body>

</html>
<style>
    tr:hover{
        cursor: pointer;
    }
</style>

```

## staffLogin.html

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <meta charset="utf-8">
    <title>staffLogin</title>
    <link href="CSS_styles/main.css" rel="stylesheet">
    <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/jquery@3.2/dist/jquery.min.js">
</script>
    <script src="https://cdn.staticfile.org/jquery-
cookie/1.4.1/jquery.cookie.min.js"></script>
</head>

<body>
    <div id="root">
        <div class="Dashboard fullscreen dark">
            <div class="Background">
                <div class="fullscreen" style="background-color: black; opacity:
1; transition: opacity 200ms ease-in-out 0s;">
                    <div class="Image fullscreen" style="background-image:
url(background.jpg);"></div>
                </div>
            </div>
            <div class="widgets fullscreen">
                <div class="container">
                    <div class="slot middleCentre">
                        <div class="widget " style="font-size: 24px;">
                            <div class="Greeting">
                                <h2>请工作人员进行身份验证</h2>
                            </div>
                        </div>
                        <div class="widget " style="font-size: 24px;">
                            <form class="Search"><input placeholder="输入身份编号"
tabindex="1" id="staff"></form>
                            <form class="Search"><input placeholder="输入密码"
tabindex="1" type="password" id="Password"></form>
                        </div>
                        <div class="widget " style="font-size: 24px;">
                            <button type="button" class="btn btn-info"
id="checkin" style="width: 300px;height: 50px;"
onclick="verify()">验证</button>

```

```

        </div>
    </div>
</div>
</div>
</div>
</div>
</body>

<script type="text/javascript ">
    function verify() {
        if (document.getElementById('staff').value == '01' &&
document.getElementById('Password').value == '123') {
            window.location.href = './admin.html';
        } else if (document.getElementById('staff').value == '02' &&
document.getElementById('Password').value == 321) {
            window.location.href = './manager.html';
        } else {
            alert("身份验证失败 ");
        }
    }
}
</script>

</html>

```

## user.html

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>user</title>
    <link rel="stylesheet"
href="https://apps.bdimg.com/libs/bootstrap/3.2.0/css/bootstrap.min.css">
    <script src="https://cdn.jsdelivr.net/npm/jquery@3.2/dist/jquery.min.js">
</script>
    <script src="https://cdn.staticfile.org/jquery-
cookie/1.4.1/jquery.cookie.min.js"></script>
    <style>
        body {
            background-image: url('background.jpg');
            background-repeat: no-repeat;
        }
    </style>
</head>

<body onload="load()" onload="clock()">
    <div class="container">
        <div class="row clearfix">
            <div class="col-md-12 column">
                <div>
                    <h1 style="text-align: center">
</h1>
                </div>
            </div>
        </div>
    </div>

```

```

    </div>
    <div class="row clearfix">
      <div class="col-md-2 column"></div>
      <div class="col-md-8 column">
        <div class="panel panel-success">
          <div class="panel-heading" style="background-color:
antiquewhite;">
            <h3 class="panel-title" style="text-align: center;font-
size:20px;">
              空调状态
            </h3>
            <h3 style="text-align: center;font-size:20px;">
              clock: <span id="clock"></span>
            </h3>
          </div>
          <div class="panel-body" style="font-size:15px; background-
color: aliceblue;">
            开关: <span id="state">关</span>
            <div class="input"></div>
          </div>
          <div class="panel-body" style="font-size:15px;background-
color: aliceblue;">
            当前风速:
            <span id="speed"></span>
          </div>
          <div class="panel-body" style="font-size:15px;background-
color: aliceblue;">
            模式:
            <span id="mode"></span>
          </div>
          <div class="panel-body" style="font-size:15px;background-
color: aliceblue;">
            当前温度:
            <span id="currentTemperature"></span>
          </div>
          <div class="panel-body" style="font-size:15px;background-
color: aliceblue;">
            目标温度:
            <span id="targetTemperature"></span>
          </div>
          <div class="panel-footer" style="font-size:15px;">
            应缴纳金额:
            <span id="fee"></span>
          </div>
        </div>
      </div>
    </div>
    <div class="col-md-2 column"></div>

  </div>
  <div class="row clearfix">
    <div class="col-md-4 column">
      </div>

    <div class="col-md-4 column">
      <div class="panel panel-success ">

```

```

<div class="panel-heading" style="background-color:
antiquewhite;">
    <h3 style="text-align: center; font-size:25px;">
        当前房间号: <span id="roomID"></span>
    </h3>
    <h3 class="panel-title" style="text-align: center; font-
size:20px;">
        遥控器
    </h3>
</div>
<div class="panel-body" style="font-size:15px;background-
color: aliceblue;">
    <button type="button" class="btn btn-lg btn-success btn-
block" style="padding: 10px 10px; background: rgb(55, 95, 226);text-align:
center;font-size:5px;" id="changeState" onclick="changeOnOff()">ON/OFF</button>
</div>
<div class="panel-body" style="font-size:15px;background-
color: aliceblue;">
    目标风速:

    <select onchange="detail(this)" id="aimspeed">
        <option value="0">低风</option>
        <option selected="selected" value="1">中风</option>
        <option value="2">高风</option>
    </select>
</div>
<div class="panel-body" style="font-size:15px;background-
color: aliceblue;">
    目标温度:
    <select onchange="detail(this)" id="aimtemp">
        <option value="18">18</option>
        <option value="19">19</option>
        <option value="20">20</option>
        <option value="21">21</option>
        <option value="22">22</option>
        <option value="23">23</option>
        <option value="24">24</option>
        <option selected="selected" value="25">25</option>
        <option value="26">26</option>
        <option value="27">27</option>
        <option value="28">28</option>
        <option value="29">29</option>
        <option value="30">30</option>
    </select>

</div>
<div class="panel-footer" style="font-size:30px;">
    <button type="button" class="btn btn-default btn-block"
style="color: white;padding: 10px 10px; background: rgb(55, 95, 226);text-align:
center;font-size:5px;" onclick="checkout()">退房</button>
</div>
</div>

</div>
<div class="col-md-4 column">
</div>
</div>
</div>

```

```

</body>

<script type="text/javascript">
    // acId = $.cookie('roomID');

    var url = location.search; //获取url中"?"符后的字符串
    if (url.indexOf("?") != -1) { //判断是否有参数
        var str = url.substr(1); //从第一个字符开始 因为第0个是?号 获取所有除问号的所有字符串
        strs = str.split("="); //用等号进行分隔 （因为知道只有一个参数 所以直接用等号进行分隔 如果有多个参数 要用&号分隔 再用等号进行分隔）
        // alert(strs[1]); //直接弹出第一个参数 （如果有多个参数 还要进行循环的）
    }

    acId = strs[1];
    //alert(" acId " + acId);

    var cTemp = 25;
    var state = document.getElementById('state').innerHTML;
    var dewind = 1;
    var deTemp = 18;
    var haveon = 0;

    function load() {
        document.getElementById("roomID").innerHTML = acId;
        // cTemp = prompt('请输入当前温度', '25');
        document.getElementById("currentTemperature").innerHTML = cTemp;
    }

    var int = self.setInterval("clock()", 1000);

    function clock() {
        var d = new Date();
        var t = d.toLocaleTimeString();
        document.getElementById("clock").innerHTML = t;
        getAirState();
    }

    function getAirState() {
        $.ajax({
            url: "http://119.3.212.133:20002/ac/state",
            type: "GET",
            data: {
                "acId": acId
            },
            success: function(result) {
                // alert("getState success");
                if (result.code == 200) {
                    if ((result.result.state == 1) || (result.result.state == 3)) {
                        document.getElementById("state").innerHTML = "开";
                    } else if (result.result.state == 2) {
                        document.getElementById("state").innerHTML = "关";
                    }
                    if (result.result.mode == 0) {
                        document.getElementById("mode").innerHTML = "制冷";
                    }
                }
            }
        });
    }

```

```

        } else if (result.result.mode == 1) {
            document.getElementById("mode").innerHTML = "制热";
        }
        if (result.result.speed == 0) {
            document.getElementById("speed").innerHTML = "低风";
        } else if (result.result.speed == 1) {
            document.getElementById("speed").innerHTML = "中风";
        } else if (result.result.speed == 2) {
            document.getElementById("speed").innerHTML = "高风";
        }
        document.getElementById("currentTemperature").innerHTML =
result.result.currentTemperature;
        document.getElementById("targetTemperature").innerHTML =
result.result.targetTemperature;
        document.getElementById("fee").innerHTML =
result.result.fee;
        cTemp = result.result.currentTemperature;
    } else alert(result.code);
},
error: function(jqXHR, textStatus, errorThrown) {
    alert("getState error");
}
})
}

function detail(sel) {
    var selectItem = sel.options[sel.selectedIndex].text;
    // alert("detail:" + selectItem);
    // document.getElementById('iden').value =selectItem;
    console.log(selectItem);
    if (selectItem == "高风") {
        // alert(selectItem);
        dewind = 2;
        changewind();
    } else if (selectItem == "中风") {
        // alert(selectItem);
        dewind = 1;
        changewind();
    } else if (selectItem == "低风") {
        // alert(selectItem);
        dewind = 0;
        changewind();
    } else {
        // alert(selectItem);
        deTemp = selectItem;
        changeTemp();
    }
    // windAndTemp(dewind,deTemp);
}

function changewind() {
$.ajax({
    url: "http://119.3.212.133:20002/ac/adjust",
    type: "GET",
    data: {
        "acId": acId,
        "dewind": dewind,
        "deTemp": -1
    }
})
}

```

```

    },
    success: function(result) {
        switch (result.code) {
            case 200:
                //alert("调整成功");
                break;
            case 500:
                alert(result.message)
                break;
            default:
                break;
        }
    },
    error: function(jqXHR, textStatus, errorThrown) {
        alert("adjust error");
    }
})
}

```

```

function changeTemp() {
    $.ajax({
        url: "http://119.3.212.133:20002/ac/adjust",
        type: "GET",
        data: {
            "acId": acId,
            "dewind": -1,
            "deTemp": deTemp
        },
        success: function(result) {
            switch (result.code) {
                case 200:
                    //alert("调整成功");
                    break;
                case 500:
                    alert(result.message)
                    break;
                default:
                    break;
            }
        },
        error: function(jqXHR, textStatus, errorThrown) {
            alert("adjust error");
        }
    })
}

```

```

function changeOnOff() {
    state = document.getElementById('state').innerHTML;
    if (state == "关") {
        turnOn();
    } else {
        turnOff();
    }
}

```

```

function turnOn() {
    if (haveon == 0) {
        cTemp = prompt('请输入当前温度', '25');
    }
}

```



```

        haveon = 1;
    }

$.ajax({
    url: "http://119.3.212.133:20002/ac/start",
    type: "GET",
    data: {
        "acId": acId,
        "cTemp": cTemp
    },
    success: function(result) {
        switch (result.code) {
            case 200:
                alert("开机成功");
                break;
            case 500:
                alert("系统未运行");
                break;
            case 501:
                alert("空调已开机");
                break;
            default:
                break;
        }
    },
    error: function(jqXHR, textStatus, errorThrown) {
        alert("error");
    }
})

}

function turnoff() {
    // alert(document.getElementById('aimespeed').value);
    document.getElementById('aimespeed').value = "1";
    document.getElementById('aimtemp').value = "25";
    // alert("恢复默认值");

$.ajax({
    url: "http://119.3.212.133:20002/ac/shutdown",
    type: "GET",
    data: {
        "acId": acId
    },
    success: function(result) {
        switch (result.code) {
            case 200:
                alert("关机成功");
                break;
            case 500:
                alert("系统未运行");
                break;
            case 501:
                alert("空调已关机");
                break;
            default:
                break;
        }
    }

```

```

    },
    error: function(jqXHR, textStatus, errorThrown) {
        alert("error");
    }
}

function checkout() {
    window.location.href = './detailList.html' + '?index=' + acid;
}
</script>

</html>

```

## userLogin.html

```

<!DOCTYPE html>
<html>

<head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <meta charset="utf-8">

    <title>staffLogin</title>
    <link href="CSS_styles/main.css" rel="stylesheet">
    <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/jquery@3.2/dist/jquery.min.js">
</script>
    <script src="https://cdn.staticfile.org/jquery-
cookie/1.4.1/jquery.cookie.min.js"></script>

</head>

<body>
    <div id="root">
        <div class="Dashboard fullscreen dark">
            <div class="Background">
                <div class="fullscreen" style="background-color: black; opacity:
1; transition: opacity 200ms ease-in-out 0s;">
                    <div class="Image fullscreen" style="background-image:
url(background.jpg);"></div>
                </div>
            </div>
            <div class="Widgets fullscreen">
                <div class="container">
                    <div class="Slot middleCentre">
                        <div class="widget " style="font-size: 24px;">
                            <div class="Greeting">
                                <h2>客户入住登记</h2>
                            </div>
                        </div>
                        <div class="widget " style="font-size: 24px;">
                            <form class="Search"><input placeholder="输入姓名"
tabindex="1" id="name" name="name"></form>

```

```
<form class="Search"><input placeholder="输入身份证"
tabindex="1" type="text" id="Password"></form>
</div>
<div class="widget " style="font-size: 24px;">
    <button type="button" class="btn btn-info"
id="checkin" style="width: 300px;height: 50px;"
        >验证</button>
    </div>
</div>
</div>
</div>
</div>
</div>
</body>

<script>
var acId;
$(document).ready(function() {
    $("#checkin").click(function() { //入住按钮点击后
        $.ajax({
            url: "http://119.3.212.133:20002/front/checkin",
            type: "GET",
            data: $("#name"),
            success: function(result) {
                // alert("success");
                if (result.code == 200) {
                    // $.cookie('roomId',result.result.acId);
                    acId = result.result.acId;
                    alert("您的房间号为: " + acId);
                    window.location.href = './user.html' + '?index=' + acId;
                }
            },
            error: function(jqXHR, textStatus, errorThrown) {
                alert("房间已满，可以去别家酒店试试看哦~");
            }
        })
    })
})
</script>
</html>
```