用户（家）的注册和登录**（周沛辰）**

数据分析模块提供的API

1. 分析用户数据**（周沛辰）**
2. 分析用户的温度分析
3. 分析用户作息时间分析
4. 分析湿度
5. 分析影音系统，什么类型音乐和电影
6. 智能设备分析**（衷睿）**
7. 设备管理模块
8. 认证功能**（衷睿）**
9. 控制设备：设备的开和关，根据分析出的数据去控制相应的设备**（余炳蔚）**
10. 添加、删除设备**（衷睿）**
11. 前端**（陈留圣）**

数据库：**（周沛辰）**

用户表（基本信息）

设备表（所有的设备，状态）

设备类型表（设备类型A1,A2,A3，满足的设备才能被添加进来）

设备使用记录表(设备id，使用时间，操作码)

环境表（时间，温度，湿度）

主人的状况表：存储时间和主人是否在家

空调，电灯，窗帘，加湿器，电视机，音响

**2020/06/20：**

操作码：//

1. AirConditioner：
2. A\_PowerOn\_当前温度\_设定温度 ，eg: A\_PowerOn\_35\_27
3. A\_PowerOff
4. A\_SetTo\_设定温度，eg: A\_SetTo\_26
5. Light
6. L\_PowerOn
7. L\_PowerOff
8. TV
9. T\_PowerOn
10. T\_PowerOff
11. Curtain
12. C\_PowerOn //拉开窗帘
13. C\_PowerOff //关上窗帘
14. Humidifier
15. H\_PowerOn\_当前湿度
16. H\_PowerOff\_当前湿度
17. Box
18. B\_PowerOn
19. B\_PowerOff

前端应该要有一个框，表示当前所有家电的状况；还要有环境框，显示环境，如时间，温度，湿度

应该还要有一个主人的状况表：存储时间和主人是否在家。应该要有一个门的系统，用于人脸识别，如前端有一个进入的输入框，可以输入0(人脸识别不通过)，1（通过）,进入后则会开始打开电灯，根据当前的环境和历史数据决定是否开空调，开加湿器，开or关窗帘。而电视机和音响则是只能根据语音识别的指令来操控的

语音和人脸识别功能

1. 语音：前端有一个输入框，输入人说的话，如打开电视，就会调用后端把电视打开，前端就会用类似弹窗显示“电视已打开”，否则显示“不明指令”

电器的操作：（空调，电灯，窗帘，加湿器，电视机，音响）

除了空调，其余的只有打开和关闭操作。空调还有调节温度的操作

前端：

1. 用户的登录注册
2. XX的家:
3. 有环境框，显示环境，如时间，温度，湿度，主人的状态，可以进行设置

//时间模拟由前端完成，暂停or继续

1. //有设备框，表示当前所有家电的状况。（开or关，空调是几度？）
2. 有语音识别接收器的输入栏，用于输入用户的语音命令。
3. 有各个家电的型号显示，状态显示和操控区，可以手动来操控设备和该设备的相应设定
4. 还要有用户的智能数据分析展示 //此处用户可以手动输入相应的预设智能设定？
5. 用户自定义的日程表
6. 用户操作的历史记录

后端用例：

1. 主人的进出家门（余）
2. 主人人脸识别，进家门后的一系列操作。如根据时间段判断是否打开电灯、窗帘，根据温度判断是否开空调，根据湿度判断是否开加湿器。
3. 主人离开家，所有的电器，如空调，电视机，电灯，音箱，加湿器自动关上。窗帘不用动。
4. 语音操控设备（余）
5. 可以用语音来打开关闭所有的设备，并且可以给空调调节温度。（我们是否要规定好语音命令？“打开。。。设备类型”）
6. 模式
7. “影音模式”，自动打开电视，关闭窗帘，关闭电灯
8. “睡眠模式”：关闭电灯，窗帘，电视，打开空调
9. 智能开关设备（该模块需要根据用户的数据分析来做智能抉择）（周）
10. 首先根据用户自定义的日程表来操控设备
11. 再根据用户的数据习惯来操控设备：
12. 根据温度自动打开**空调**（自动打开）

//打开后根据使用的时长，智能关闭空调

1. 根据湿度自动打开or关闭**加湿器**
2. 根据时间段智能关闭or打开**窗帘**
3. 用户自定义的日程表（周）
4. 可以设置相应的时间段对设备的操作

//时间都是以分钟为单位

//前端显示的时间都是相对当天的，显示几分钟，如3:00则是180，14:00则是14\*60=840

//时间间隔timeInterval：30，表示时间间隔是30分钟

//其余的电灯，音响，电视机则没有智能功能

1. 操控设备（衷）
2. 前端用户可以手动输入，打开或关闭所有的电器，还可以进行空调的温度调节

//写入operation

设备类{

State

Id

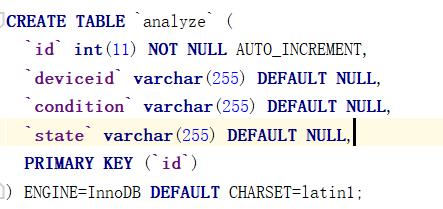
}

1. 设备管理（衷）
2. 用户设备的添加和删除功能。（根据数据库已有的型号进行添加数据）
3. 添加设备时，前端模拟认证
4. 对所有设备的开和关，对空调的温度调节
5. 设备的数据分析（周）
6. 分析用户在几度以上/以下的时候喜欢开空调，并且是开几度？
7. 分析用户在湿度为多少的时候喜欢开、关加湿器？
8. 分析用户在几点会去打开or关闭窗帘？（这个需要分析吗？还是由用户可以自动设置几点到几点之间进行拉、关窗帘）

//需要允许用户进行数据预设吗？

//数据表怎么设计？

//每次控制完设备后，就要调用该数据分析模块，去更新该数据表analyze



// 空调：condition就表示几度时用户会去开,state是设定为几度。这里空调我们分为两种，分别是20度以上和以下，用户判定是冷气还是暖气

加湿器：condition表示湿度为多少用户会去操作，state为0表示用户的关操作， 1表示用户的开操作

窗帘：condition是几点，state为0表示用户的关操作， 1表示用户的开操作

1. 家的环境和主人的状态（余）
2. 温度调节的接口
3. 湿度调节的接口
4. 主人的状态接口

//时间，日期，温度，湿度

//在家、不在家

1. 用户的登录和注册功能（周）

数据库

1. environment表添加ownerid,主人的状态
2. device表，空调的状态可以是0~255，0是关闭，其他的代表温度
3. 用户数据分析表（id,ownerid,deviceid,）
4. 日程表（id,ownerid,time,opercode）

操作码

时间的表示

前端传环境参数给后端，后端要先判断该时间段内有没有日程，有的话则调用操控设备的接口，该接口里调用完设备后，还要调用分析用户数据的接口。然后根据分析的数据判断该环境下是否需要智能开启设备（空调，加湿器，窗帘）。

主人一回家，就根据时间判断是否打开电灯（需要根据时间判断？），然后调用智能自动操控设备的模块接口

Device表新增userId，表明该设备是属于哪个用户的。数据库多处已更新

Id是自增长，我在本地的device里已经创建了几条数据，

日程表开和关是分开的