浙江大学

本科实验报告

课程名称：	B / S 体系软件设计
姓　　名：	沈韵泓
学　　院：	计算机科学与技术学院
系：	计算机科学与技术系
专　　业：	软件工程
学　　号：	3200104392
指导教师：	胡晓军

2022 年 10 月 23 日

浙江大学实验报告

课程名称：B / S 体系软件设计 实验类型：个人上机实验

实验项目名称：智能家居管理系统

学生姓名：沈韵泓 专业：软件工程 学号：3200104392

同组学生姓名：（无） 指导老师：胡晓军

实验地点：曹光彪西楼 503 实验日期：2022 年 10 月 31 日

目录

智能家居管理系统设计文档

版本	编写人员	最后修改时间
1.0	沈韵泓 3200104392	2022/10/26

1 设计概述

1.1 任务和目标

本项目旨在实现一个智能家居管理网站，用户可以在注册、登录后使用该网站创建场所、房间与设备，并通过可视化的方式实时对家居设备进行查看与管理。此外，本平台界面美观，且同时适配 PC 端与移动端。

1.2 运行环境

本项目采用前后端分离架构，前端技术栈为 HTML + CSS + JavaScript，同时采用 Bootstrap 框架实现响应式布局。后端技术栈为 Flask + MySQL。

服务器部署环境要求如下：

- Bootstrp v3.4.1 及以上
- Python 3.10.8 及以上
- Pip 22.3 及以上
- Flask 2.2.2 及以上
- Flask SQLAlchemy 1.4.42 及以上
- Werkzeug 2.2.2 及以上
- MySQL 8.0.31 及以上

用户端运行环境如下：

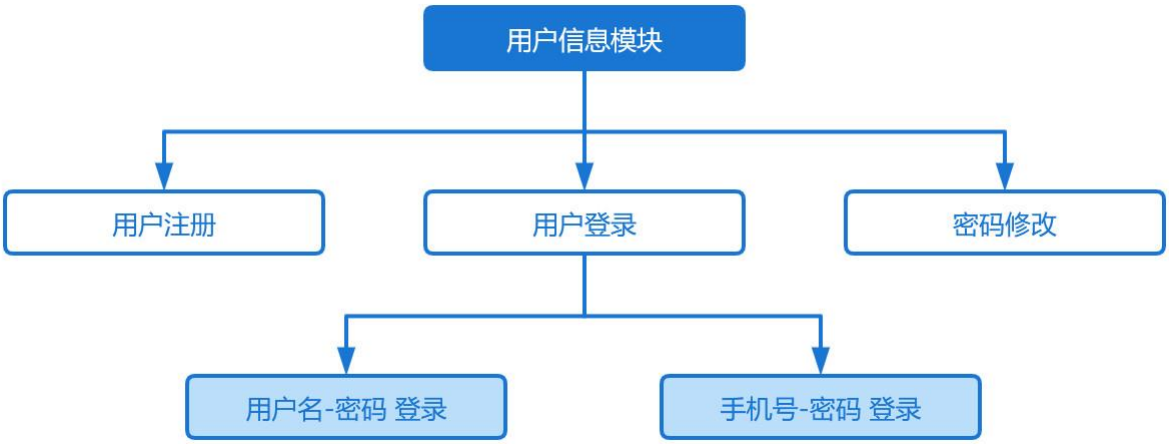
PC 端与移动端均适配各种主流浏览器。

PC 端操作系统：Windows7, Windows10, MACOS。

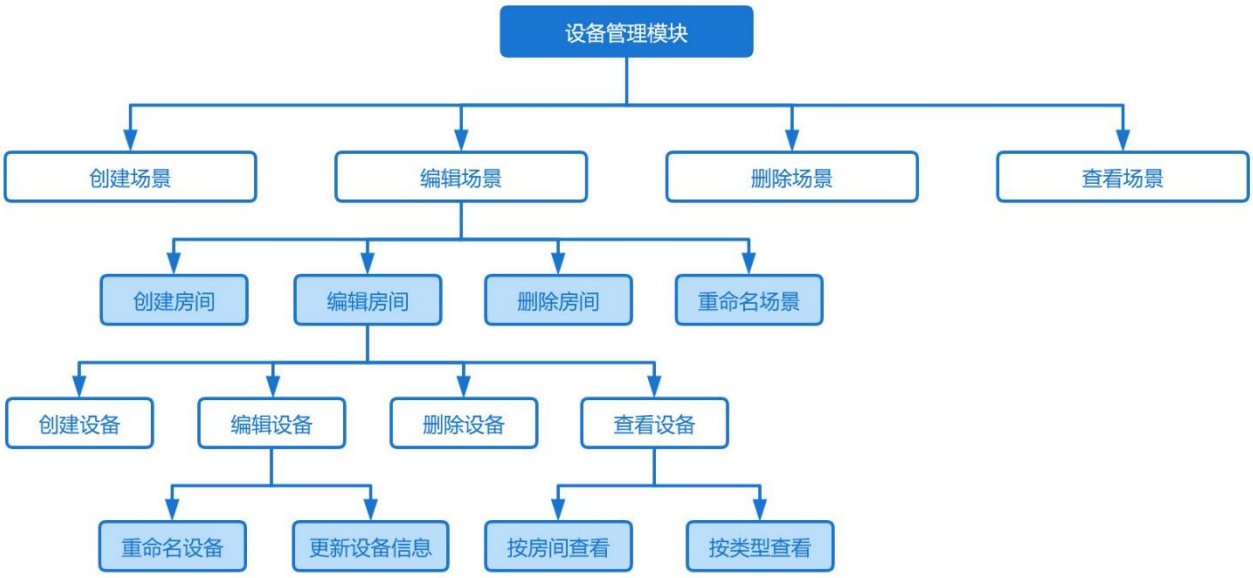
移动端操作系统：Android, Harmony。

2 功能模块与层次结构设计

2.1 用户信息模块

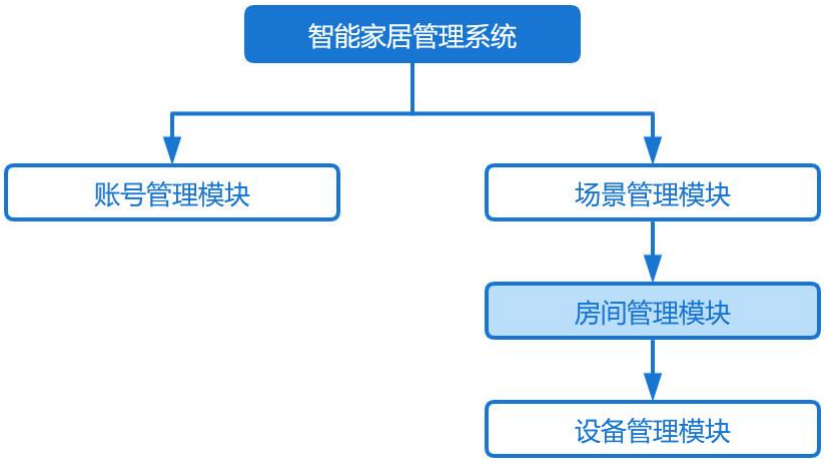


2.2 设备管理模块



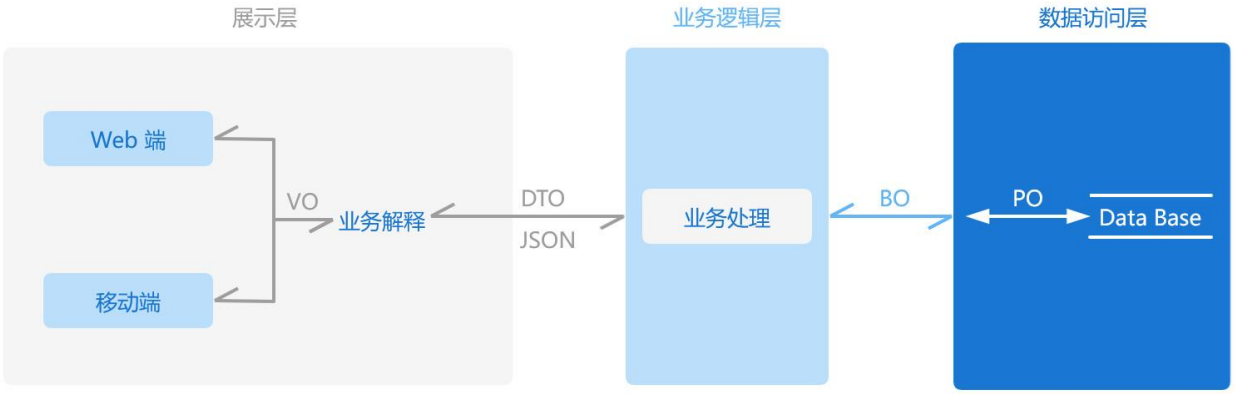
2.3 用户端项目层次结构

通过对题干进行分析，我们将智能家居管理系统拆分为以下四个模块：



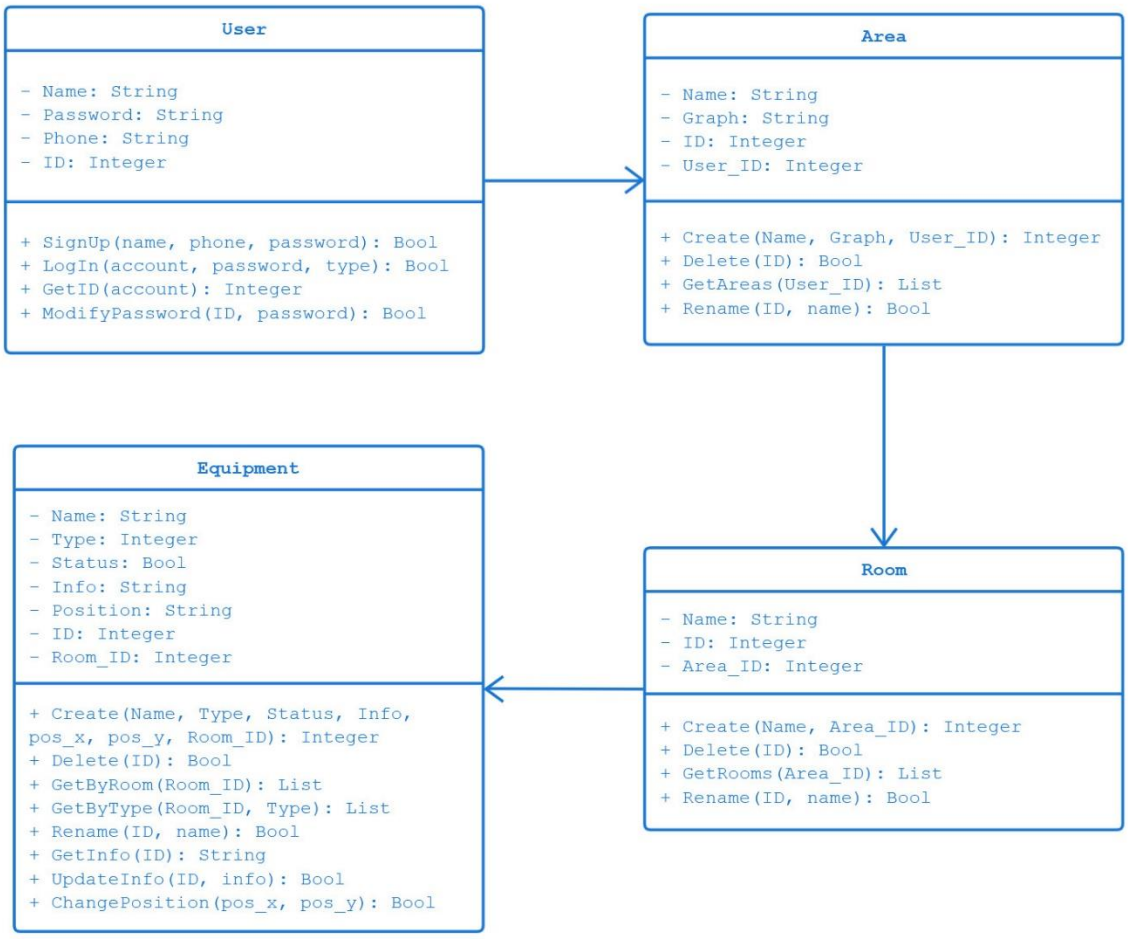
- 账号管理模块
  - 支持用户注册登录
  - 支持用户修改头像等账号信息
- 场景管理模块
  - 支持以卡片形式对账号下属场景的显示
  - 支持场景创建与删除
  - 支持对已创建场景进行重命名
- 房间管理模块
  - 支持以列表形式对场景下属房间及其设备进行显示
  - 支持房间创建与删除
  - 支持对已创建房间进行重命名
- 设备管理模块
  - 支持设备的创建与删除
  - 支持修改与查询已创建设备信息
  - 支持以颜色信息反馈设备运行状态
  - 支持修改设备位置信息

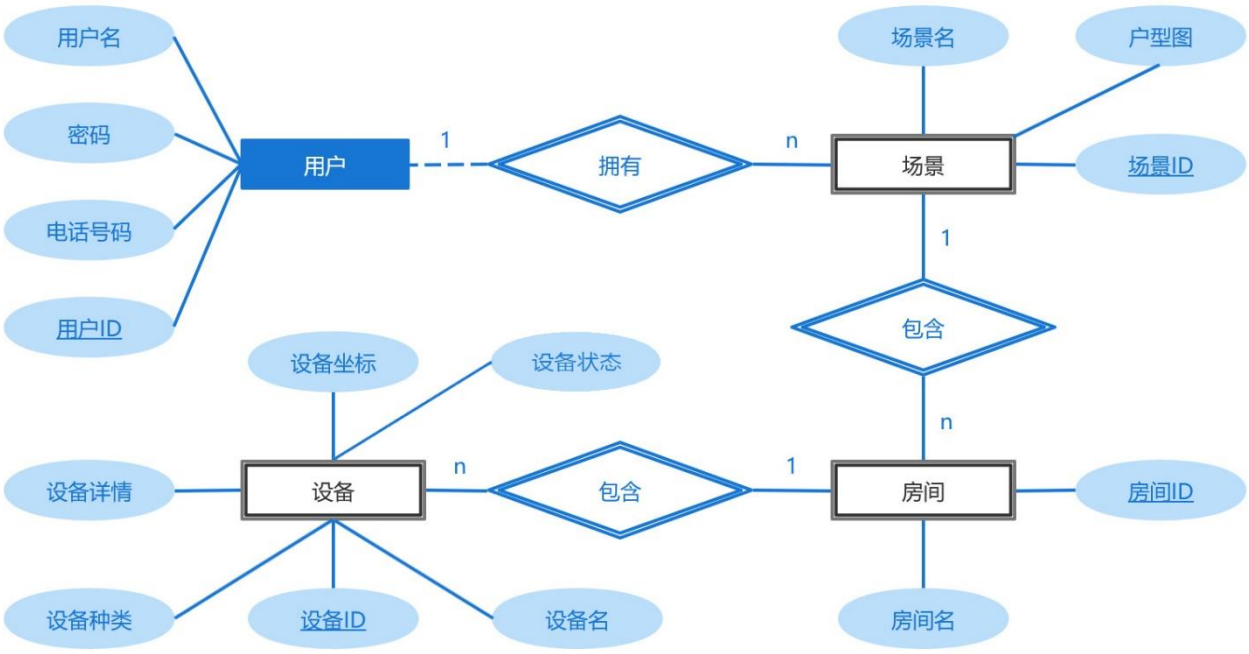
2.4 服务端项目层次结构



2.5 系统整体架构

3 类图与 ER 图


User
- Name: String
- Password: String
- Phone: String
- ID: Integer
+ SignUp(name, phone, password): Bool
+ LogIn(account, password, type): Bool
+ GetID(account): Integer
+ ModifyPassword(ID, password): Bool
Area
- Name: String
- Graph: String
- ID: Integer
- User_ID: Integer
+ Create(Name, Graph, User_ID): Integer
+ Delete(ID): Bool
+ GetAreas(User_ID): List
+ Rename(ID, name): Bool
Equipment
- Name: String
- Type: Integer
- Status: Bool
- Info: String
- Position: String
- ID: Integer
- Room_ID: Integer
+ Create(Name, Type, Status, Info, pos_x, pos_y, Room_ID): Integer
+ Delete(ID): Bool
+ GetByRoom(Room_ID): List
+ GetByType(Room_ID, Type): List
+ Rename(ID, name): Bool
+ GetInfo(ID): String
+ UpdateInfo(ID, info): Bool
+ ChangePosition(pos_x, pos_y): Bool
Room
- Name: String
- ID: Integer
- Area_ID: Integer
+ Create(Name, Area_ID): Integer
+ Delete(ID): Bool
+ GetRooms(Area_ID): List
+ Rename(ID, name): Bool


用户名
密码
电话号码
用户ID
用户
1
拥有
n
场景名
户型图
场景
场景ID
1
包含
n
设备坐标
设备状态
设备详情
设备
n
包含
1
房间
房间ID
设备种类
设备ID
设备名
房间名

4 数据字典

4.1 User 表

字段名	类型	描述	备注
Name	varchar(15)	用户名	NOT_NULL;UNI
Password	varchar(15)	密码	NOT_NULL
Phone	char(11)	手机号	NOT_NULL;UNI
ID	int(10)	用户 ID	PRI; AUTO_INC

4.2 Area 表

字段名	类型	描述	备注
Name	varchar(15)	场景名	NOT_NULL
Graph	varchar(50)	户型图路径	NOT_NULL
ID	int(10)	场景 ID	PRI; AUTO_INC
User_ID	int(10)	用户 ID	NOT_NULL;FOR

4.3 Room 表

字段名	类型	描述	备注
Name	varchar(15)	房间名	NOT_NULL
ID	int(10)	房间 ID	PRI; AUTO_INC
Area_ID	int(10)	场景 ID	NOT_NULL;FOR

4.4 Equipment 表

字段名	类型	描述	备注
Name	varchar(15)	设备名	NOT_NULL
Type	Int(2)	设备类型	NOT_NULL
Status	Bool	设备状态 On/Off	NOT_NULL
Info	varchar(50)	设备详情	
Position	varchar(20)	设备定位	NOT_NULL
ID	int(10)	设备 ID	PRI; AUTO_INC
Room_ID	int(10)	房间 ID	NOT_NULL;FOR

5 接口设计

[说明] 除 /signup 与 /login 外，所有请求都需要通过 user_id 验证登陆状态。

5.1 用户注册

URL	/signup
请求方式	POST

请求格式	{
	“user_name”: 用户名,
	“password” : 密码,
	“phone” : 手机号
	}

响应格式	{
	“code”: 状态码
	}

备注	返回的状态码有以下四种可能： ● 100：SUCCESS ● 200：FAIL ● 201：用户名已存在 ● 202：手机号已存在

5.2 用户登录

URL	/login
请求方式	POST

请求格式	{
	“account” ：账号，
	“password” ：密码
	}

响应格式	{
	“user_id”：用户 ID，
	“code” ：状态码
	}

备注	账号可以为：用户名/手机号 返回的状态码有以下三种可能： ● 100：SUCCESS ● 200：FAIL ● 203：用户不存在

5.3 用户注销

URL	/logout
请求方式	GET
请求格式	{ “user_id”：用户 ID }
响应格式	{ “code”：状态码 }
备注	返回的状态码有以下两种可能： ● 100：SUCCESS ● 200：FAIL

5.4 创建场景

URL	/newarea
请求方式	POST

请求格式	{
	“area_name”: 场景名,
	“graph” : 户型图,
	“user_id” : 用户 ID
	}

响应格式	{
	“area_id” : 场景 ID,
	“code” : 状态码
	}

备注	返回的状态码有以下两种可能： 100：SUCCESS / 200：FAIL

5.5 删除场景

URL	/delarea
请求方式	POST

请求格式	{
	“area_id”: 场景 ID,
	“user_id”: 用户 ID
	}

响应格式	{ “code”: 状态码 }
备注	返回的状态码有以下两种可能： ● 100：SUCCESS ● 200：FAIL ● 204：场景不存在

5.6 重命名场景

URL	/rearea
请求方式	POST

请求格式	{
	“area_id” ：场景 ID,
	“area_name” ：新场景名,
	“user_id” ：用户 ID
	}

响应格式	{ “code”：状态码 }
备注	返回的状态码有以下三种可能： ● 100：SUCCESS ● 200：FAIL ● 204：场景不存在

5.7 获取场景列表

URL	/getarealist
请求方式	POST
请求格式	{ “user_id”: 用户 ID }

响应格式	{
	“code” : 状态码,
	“n_area” : 总场景数,
	“arealist”:
	[{
	“area_id” : 场景 ID,
	“area_name” : 场景名
	}, ...
	]
	}

备注	返回的状态码有以下两种可能： ● 100：SUCCESS ● 200：FAIL

5.8 创建房间

URL	/newroom
请求方式	POST

请求格式	{
	“room_name”: 场景名,
	“area_id” : 场景 ID,
	“user_id” : 用户 ID
	}

响应格式	{
	“room_id” : 场景 ID,
	“code” : 状态码
	}

备注	返回的状态码有以下两种可能： 100：SUCCESS / 200：FAIL

5.9 删除房间

URL	/delroom
请求方式	POST

请求格式	{
	“room_id”: 场景 ID,
	“user_id”: 用户 ID
	}

响应格式	{ “code”: 状态码 }
备注	返回的状态码有以下三种可能： ● 100：SUCCESS ● 200：FAIL ● 205：房间不存在

5.10 重命名房间

URL	/reroom
请求方式	POST

请求格式	{
	“room_id” : 房间 ID,
	“room_name” : 新房间名,
	“user_id” : 用户 ID
	}

响应格式	{ “code”: 状态码 }
备注	返回的状态码有以下三种可能： ● 100：SUCCESS ● 200：FAIL ● 205：房间不存在

5.11 获取房间列表

URL	/getroomlist
请求方式	POST

请求格式	{
	“area_id” ：场景 ID,
	“user_id” ：用户 ID
	}

响应格式	{
	“code” ：状态码,
	“n_room” ：总房间数,
	“roomlist”:
	[{
	“room_id” ：房间 ID,
	“room_name” ：房间名
	}, ...
	]
	}

备注	返回的状态码有以下三种可能： ● 100：SUCCESS ● 200：FAIL ● 204：场景不存在

5.12 创建设备

URL	/newequip
请求方式	POST

请求格式	{
	“equip_name”: 设备名,
	“equip_type”: 设备类型,
	“equip_status”: 设备状态,
	“equip_info”: 设备详情,
	“pos_x”: X 坐标,
	“pos_y”: Y 坐标,
	“room_id”: 房间 ID,
	“user_id”: 用户 ID
	}

响应格式	{
	“equip_id” : 设备 ID,
	“code” : 状态码
	}

备注	返回的状态码有以下两种可能： ● 100：SUCCESS ● 200：FAIL

5.13 删除设备

URL	/delequip
请求方式	POST

请求格式	{
	“equip_id” : 设备 ID,
	“user_id”: 用户 ID
	}

响应格式	{ “code”: 状态码 }
备注	返回的状态码有以下两种可能： ● 100：SUCCESS ● 200：FAIL

5.14 重命名设备

URL	/reequip
请求方式	POST

请求格式	{
	“equip_id” : 设备 ID,
	“equip_name” : 新设备名,
	“user_id” : 用户 ID
	}

响应格式	{ “code”: 状态码 }
备注	返回的状态码有以下两种可能： ● 100：SUCCESS ● 200：FAIL ● 206：设备不存在

5.15 获取设备状态

URL	/getequipstatus
请求方式	POST

请求格式	{
	“equip_id” ：设备 ID,
	“user_id” ：用户 ID
	}

响应格式	{
	“code” ：状态码,
	“equip_id” ：设备 ID,
	“equip_status” ：设备状态
	}

备注	返回的状态码有以下三种可能： ● 100：SUCCESS ● 200：FAIL ● 206：设备不存在

5.16 更改设备状态

URL	/modequipstatus
请求方式	POST

请求格式	{
	“equip_id” : 设备 ID,
	“equip_status” : 新设备状态 ,
	“user_id” : 用户 ID
	}

响应格式	{ “code”: 状态码 }
备注	返回的状态码有以下三种可能： ● 100：SUCCESS ● 200：FAIL ● 206：设备不存在

5.17 获取设备信息

URL	/getequipinfo
请求方式	POST

请求格式	{
	“equip_id” ：设备 ID,
	“user_id” ：用户 ID
	}

响应格式	{
	“code” ：状态码,
	“equip_id” ：设备 ID,
	“equip_info” ：设备信息
	}

备注	返回的状态码有以下三种可能： ● 100：SUCCESS ● 200：FAIL ● 206：设备不存在

5.18 更改设备信息

URL	/modequipinfo
请求方式	POST

请求格式	{
	“equip_id” ：设备 ID,
	“equip_info” ：新设备信息 ，
	“user_id” ：用户 ID
	}

响应格式	{ “code”：状态码 }
备注	返回的状态码有以下三种可能： ● 100：SUCCESS ● 200：FAIL ● 206：设备不存在

5.19 获取设备列表

URL	/getequiplist
请求方式	POST

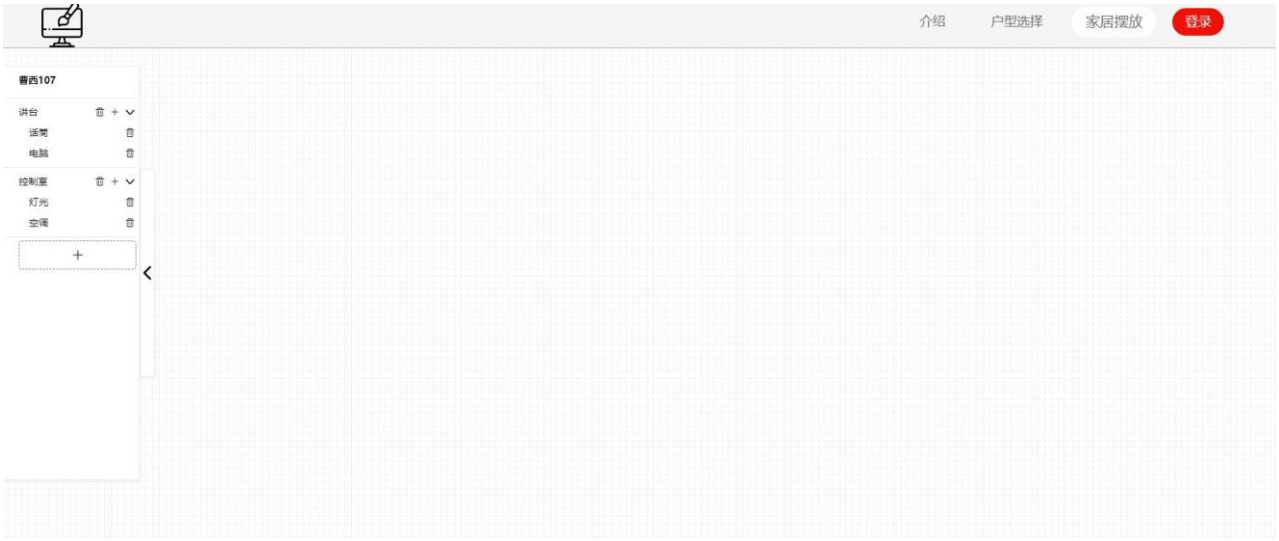
请求格式	{
	“room_id” : 房间 ID,
	“equip_type”: 要求的设备类型,
	“equip_status”: 要求的设备状态,
	“user_id” : 用户 ID
	}

响应格式	{
	“code” : 状态码,
	“n_equip” : 总设备数,
	“equiplist”:
	[{
	“equip_id” : 设备 ID,
	“equip_name” : 设备名,
	“equip_type” : 设备类型,
	“equip_status”: 设备状态,
	“equip_info” : 设备信息,
	“pos_x” : 设备横坐标,
	“pos_y” : 设备纵坐标
	}, ...
	]
	}

备注	设备类型为空时，返回所有类型的设备； 设备状态为空时，返回所有状态的设备， 返回的状态码有以下两种可能： 100：SUCCESS / 200：FAIL

6 界面原型

主界面原型如下图所示：



本平台计划将主要服务界面划分为顶栏、侧边栏与主要展示区三部分：

- “顶栏”实现了不同功能（标签页）的切换，主要包括：网站主页入口、场景（户型）管理入口与具体智能家居设备查看与摆放入口，用户登陆前提供登录入口、登录状态下则显示登录信息。
- “侧边栏”以房间为单位显示所有已创建的智能家居设备列表，并提供创建入口。
- “展示区”则将在之后显示户型图及各家居设备在该平面图上的位置，同时通过图标颜色实时反馈该设备当前的运行状态（ON/OFF）。此外，用户可以通过鼠标指针悬停来查看该设备的详细信息。

7 预计开发时间轴

2022/09/23-2022/09/30 项目需求分析

2022/09/31-2022/10/14 开发技术选型与学习

2022/10/15-2022/10/22 界面原型与接口设计

2022/10/23-2022/10/30 设计文档撰写、数据库建表与实现

2022/10/31-2022/11/07 服务端代码实现、接口文档撰写

2022/11/08-2022/11/22 前端页面撰写

2022/11/23-2022/11/30 前后端联调

2022/12/01-2022/12/07 测试文档撰写与测试

2022/12/08-2022/12/15 完成用户手册并上线

8 附录

8.1 数据字典中缩写注释

- PRI：主键
- NOT_NULL：非空
- AUTO_INC：自增
- FOR：存在外键约束

8.2 状态码设计

- 100：操作成功
- 200：FAIL，通用错误代码
- 201：用户名已存在
- 202：手机号已存在
- 203：用户不存在
- 204：场景不存在
- 205：房间不存在
- 206：设备不存在