

## 快速开发指南

SysX

成都双链科技有限责任公司

2019.3

# 变更历史

| 日期 | 变更类型 | 变更人 | 说明 |
| --- | --- | --- | --- |
| 2019年3月11日 | 创建 | 张喜来 | 创建初期版本 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 前 言

本手册分为三个部分，第一个部分是讲述如何建模，并且在5分钟内生成一个基本系统并部署，产品经理和开发均可阅读，第二部分针对后端开发人员定制系统后端，第三部分针对前端开发人员定制前端系统。每个部分都讲述原理，工具，方法，过程，验证和最佳实践。

第一部分为建模：建模的过程主要是通过客户或者产品经理提供领域模型和界面原型，使用XML语法，建立不同类型的字段和对象之间的关联。

第二部分为后端定制：后端开发通过在特定的目录中使用或者加入新的Java类，这些Java类继承生成的代码来扩展系统的后端功能。

第三部分为后端定制：前端端开发通过在特定的目录中使用或者加入新的React类，这些React类继承生成的代码来扩展系统的界面功能。

# 第一部分：建模

## 建模的原理

建模的过程主要是通过特定格式的XML文件描述对象之间的关系和自身属性的过程，系统通过这些关系和属性进行关系推演，发现基本的逻辑和业务。这些信息会在后续的处理过程中，生成前端界面，后台服务，数据库规格定义和用于测试和演示的数据。

## 建模的工具

理论上，编写XML文件任何文本编辑器都可以，我们推荐使用微软公司出品的vs code，该软件在Windows， Mac，Linux下均可以使用。

该软件的主界面如下：

## 建模的方法

在开始建模之前，一定要对业务有一个比较深入的认识，对于建模，了解高层次的分析领域模型和有哪些角色使用这个系统是必须的。界面原型是比较好的输入，可以帮助产品人员确定相关的字段。一旦确定如上事项，就可以进入编写XML文件的工作。

## 建模的过程

### 认识XML文件

XML文件是一种W3C规范的国际通用标记语言，利于计算机处理和人工编写。XML有很多高级特性，为简单起见在本手册里面只是介绍了这种语言的一个子集，便于沟通。

我们用到的是主要有属性名，属性值。

第一部分：模板

建设为his.xml, 文件名为系统名称，为英文，小写，无需下划线

<?xml version='1.0' ?>

<root **chinese\_name**="医生排班系统" **english\_name**="Doctor Scheduling System"**org**="doublechaintech" >

</root>

其中root是标签，chinese\_name, english\_name, org 是属性名称，医生排班系统，Doctor Scheduling System，doublechaintech是属性值。

| 序号 | 属性名称 | 描述 | 参考 |
| --- | --- | --- | --- |
| 1 | chinese\_name | 中文名称 | 将会显示在界面首页上 |
| 2 | english\_name | 英文名称 | 显示在英文版的界面首页上 |
| 3 | org | 开发公司的名称 | 比如doublechaintech，后台代码会显示为包名 com.doublechaintech |
|  |  |  |  |

第二部分：建一个简单的系统

我们在模型中增加了一个hospital（医院）对象，这个对象是整个系统的树根，其他的对象直接或者间接会依赖这个对象存在。

在两个root标签之间，可以输入内容。

<?xml version='1.0' ?>

<root **chinese\_name**="医生排班系统" **english\_name**="Doctor Scheduling System" **org**="doublechaintech" >

<hospital

name="上和医院|[1,200]"

address="毕升路22号"

telephone="028-9123123"

/>

</root>

hospital为标签，这里的所有标签都可以自己定义。规则是，小写加上下划线分割，比如small\_hispital。对应到数据库，会对应一张表。

name为属性名，可以自动定义，规则是，小写加上下划线分割，比如hispital\_name，"上和医院"为属性值，[1,200]标明了这个字段的最大和最小长度。如果定义将在附表中体现

标签和属性名最好使用无拼写错误的英文，这样很容易翻译为中文。拼写错误的单词，系统会给出警告。后续章节会指示如何发现拼写错误。

有了医院以后，我们希望管理医院里医生信息。以下的例子代码医院里面可以管理很多医生，医生在医院里面工作(这里没有考虑医生在多个医院工作的情况）。

黑体部分建立了医生和医院的关联，其中$(hospital)中的hospital必须是已经定义的对象。名称可以改为其他。

有了这个文本文件，我们可以根据指令上传文件，执行相关命令，就可以生成系统了。

<?xml version='1.0' ?>

<root **chinese\_name**="医生排班系统" **english\_name**="Doctor Scheduling System"**org**="doublechaintech" >

<hospital

name="上和医院|[1,200]"

address="毕升路22号"

telephone="028-9123123"

/>

<doctor

name="魏松全"

shot\_image="snap.shot-400-300-red.jpg"

**hospital="$(hospital)"**

update\_time="updateTime()"

/>

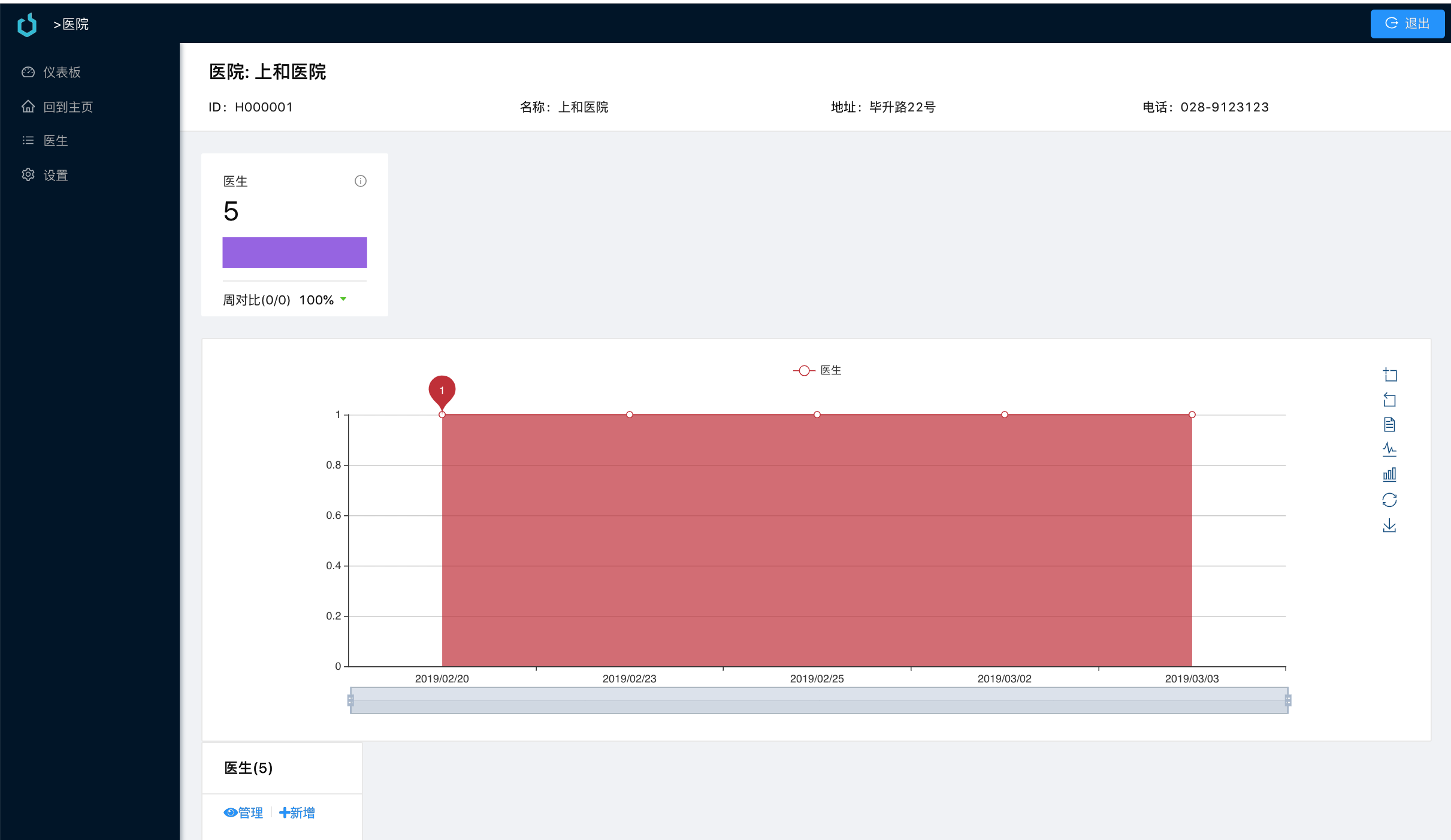
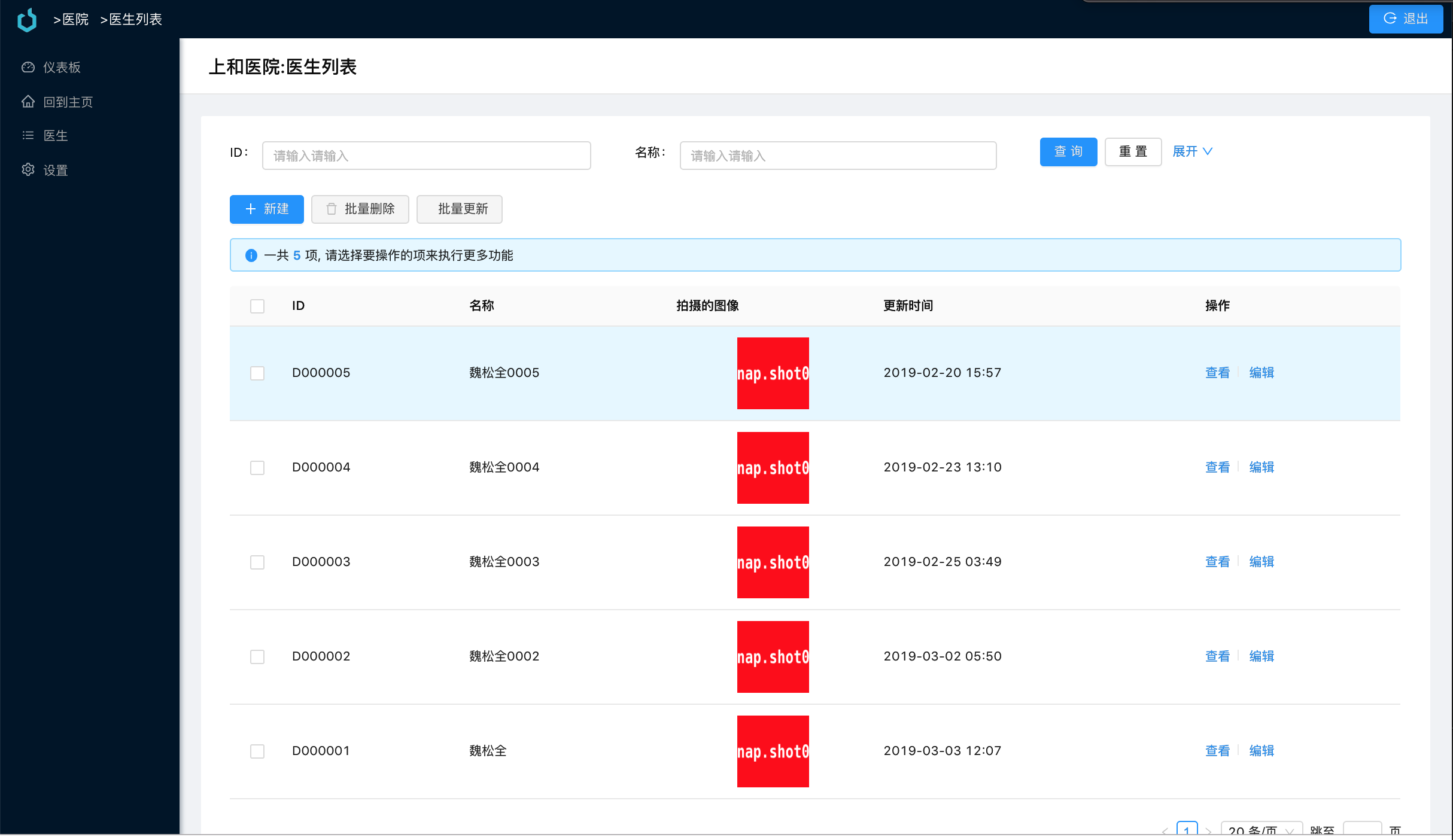
</root>

登录界面

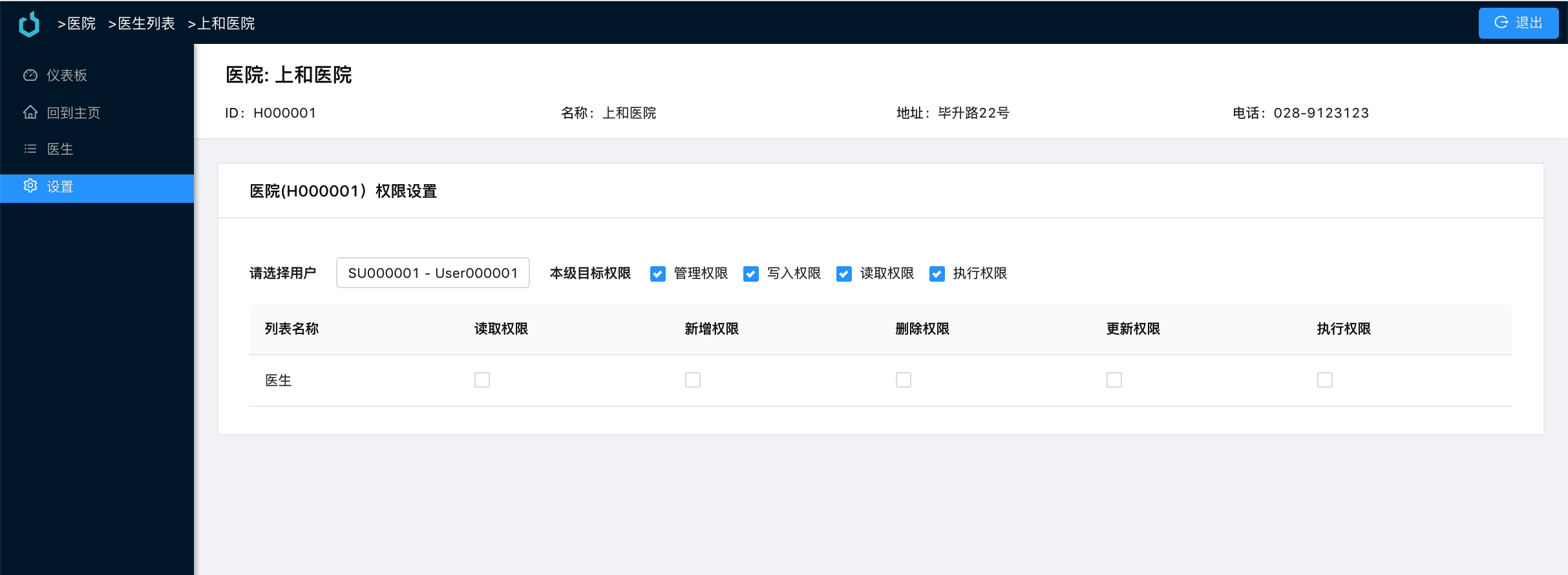
登录之后可以选择自己的app

进入主界面，可以看见生成了趋势图。

点开医生列表，可以查看医生列表。通过这个界面，基本的增删改查导航都做好了

每个字段按照定义顺序生成，ID是系统生成，除了文本字段以为，如果产品经理输入了 shot\_image="snap.shot-400-300-red.jpg", 系统就自动推断为是一幅图片，这样在展示的时候，系统就自动展示为图片，自动生成上传下载的代码，400-300-red标明了尺寸和底色。更多规则请参考附表。

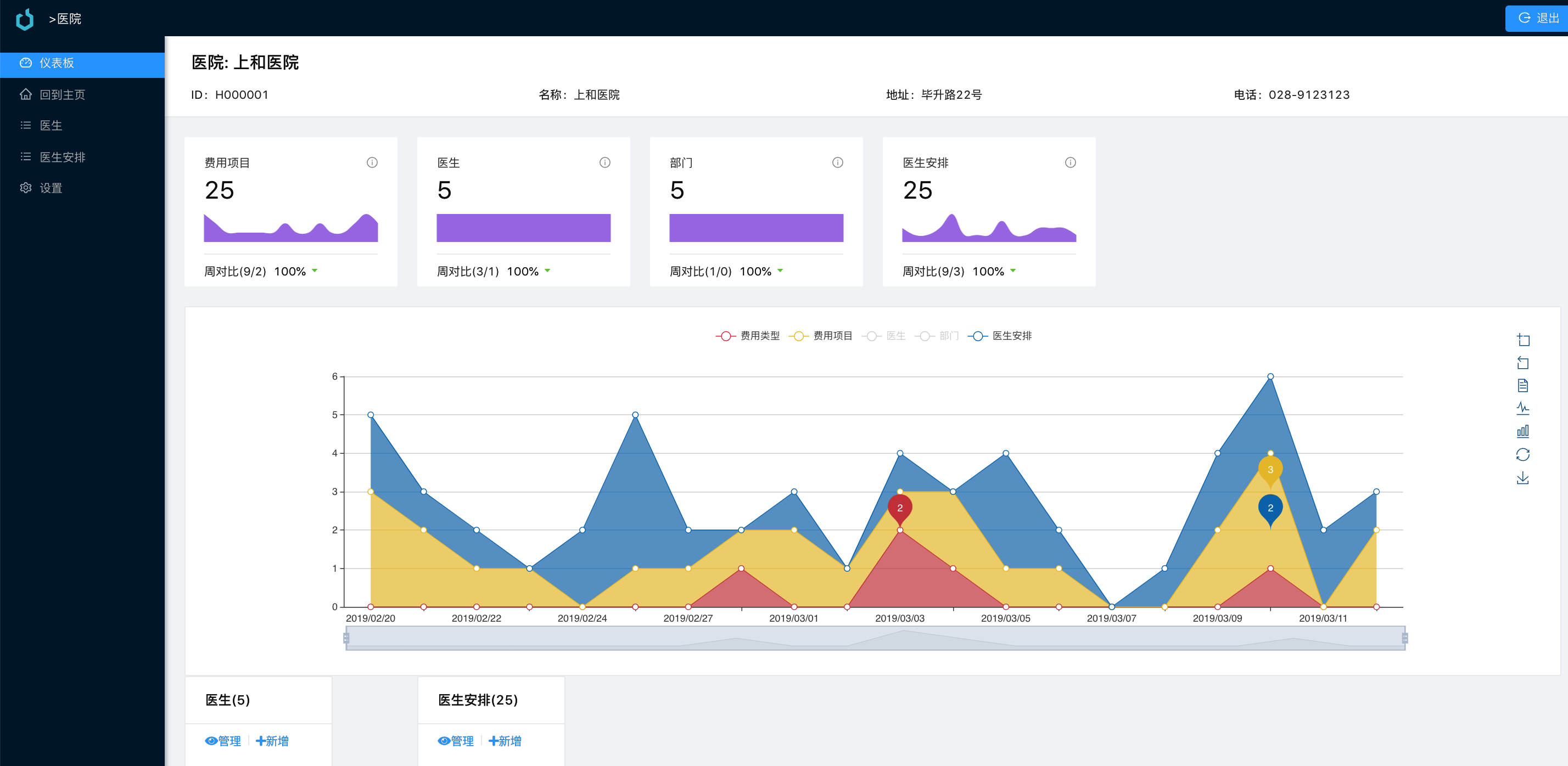
还可以设置哪些用户可以访问医生列表

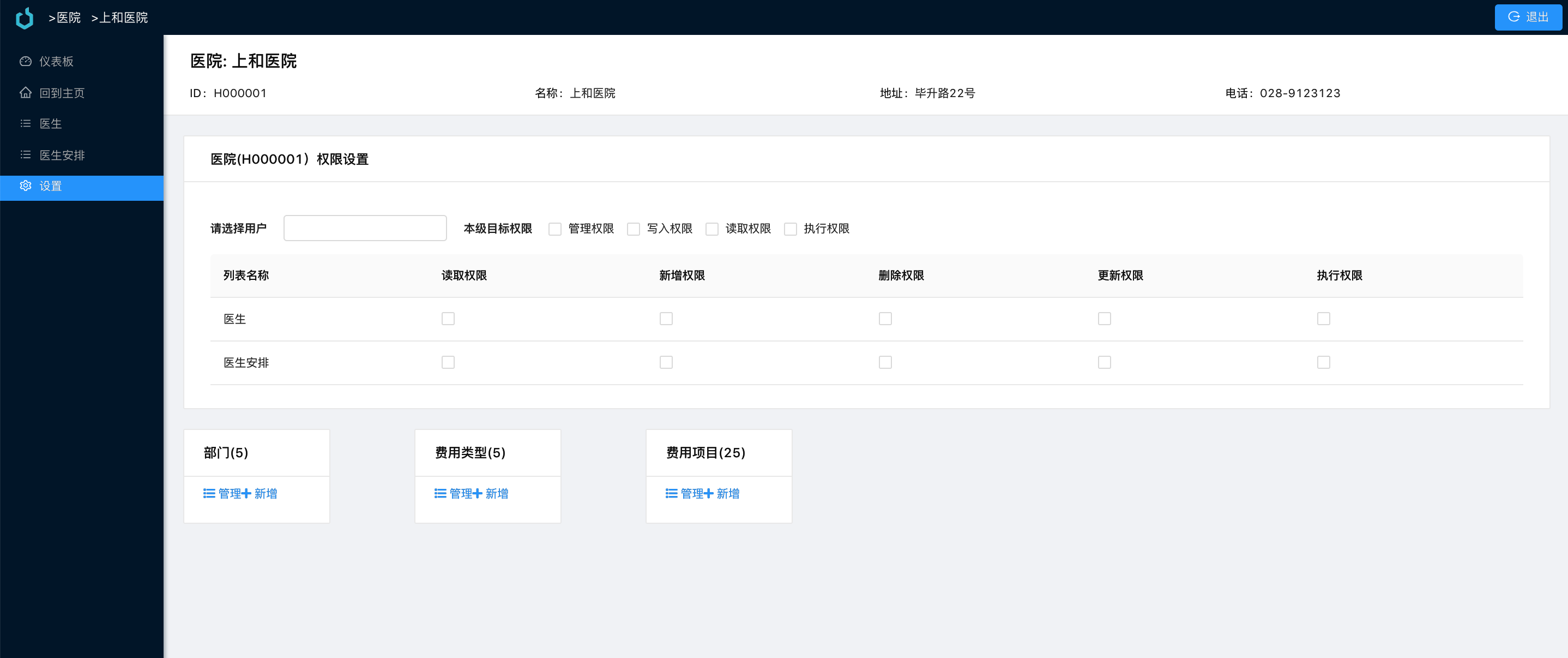


我们大约10行的文本，经过系统一系列处理，就可以生成前端，后端代码，并且配置好了演示和测试数据。

第三部分：建一个排班系统

这个例子我们将增加一些模型，这些模型增加完成以后，前端系统看起来是这样的。

系统自动生成了相关图表

权限管理增加了”医生安排“ 相应项目

我们会在后面解释为何能生成这些界面。

分析

医生排班系统的核心是这么一句话:

某个医生在某个科室在上午，下午或者晚上值班，每个排班可能有若干个号，用完为止。

记下这个事，我们需要如下的结构

<doctor\_schedule

name="2019年3月11日魏松全在内分泌科坐班收诊疗费,每个10"

doctor="$(doctor)"

schedule\_date="2019-3-11"

period="上午|下午|夜班"

department="$(department)"

available="20"

price="$123.99"

expense\_type="$(expense\_type)"

create\_time="createTime()"

update\_time="updateTime()"

hospital=“$(hospital)"

/>

里面提及了 doctor, department, expense\_type, 没有建立，我们让这些对象都存在于一个医院下。

在上述例子中，如果我们不需要，expense\_type等对象显示到主仪表盘上，我们加上了一个特性 \_features=“setting”。加入了setting，就会出现在系统设置区域内。请参考下图。

<expense\_type

name="诊疗费|治疗费|检查费"

helper\_chars="zlf|zlf|jcf"

status="正常|停用"

hospital="$(hospital)"

description="text()"

update\_time="updateTime()"

\_features="setting"

/>

<expense\_item

name="专家诊疗费|血常规|煎药费"

price="$99.00"

expense\_type="$(expense\_type)"

hospital="$(hospital)"

update\_time="updateTime()"

\_features="setting"

/>

<doctor

name="魏松全"

shot\_image="snap.shot-400-300-red.jpg"

hospital="$(hospital)"

update\_time="updateTime()"

/>

<department

name="放射科"

hospital="$(hospital)"

update\_time="updateTime()"

\_features="setting"

/>

<doctor\_assignment

name="魏松全在内分泌科室上"

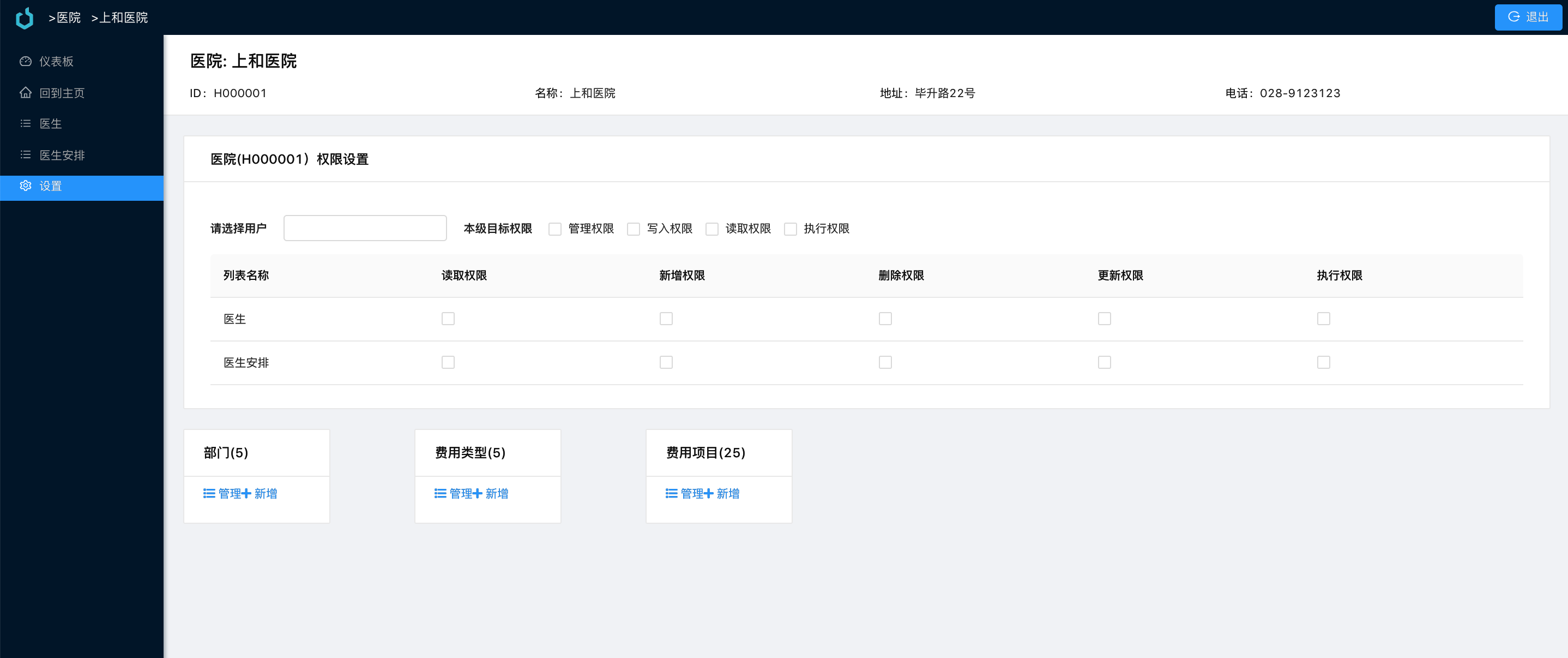
doctor="$(doctor)"

department="$(department)"

update\_time="updateTime()"

\_features="setting"

/>



自动类型推断附表

| 例子 | 类型 | 说明 | 影响 |
| --- | --- | --- | --- |
| 简单 | 字符串 | 长度内部约束 | 约束检查 |
| 简单|[1,20] | 字符串 | 长度从1到20 | 约束检查 |
| text() | 字符串 | 长字符串，约束受WEB服务限制 | 约束检查 |
| 简单|复杂 | 字符串 | 长度内部约束 | 生成数据重复简单和复杂 |
| $(hospital) | 对象引用 | 表示这个对象”附着“在另外一个对象之上。 | 对象管理，导航 |
| 11 | 整形数字 | 表示最大值为11 |  |
| 3|11 | 整形数字 | 表示最大值为11，最小值为3 |  |
| $10.99 | 金钱类型 | 金钱类型 | 约束检查，最多为10.99元，界面会显示红色，Java内部为BigDecimal |
| 10.99 | 定点小数 | 精度为2的数字 | 会自动格式化 |
| createTime() | 日期时间类型 | 创建时自动赋值 | 无需开发人员管理 |
| updateTime() | 日期时间类型 | 更新时自动赋值 | 无需开发人员管理 |

特性表(\_features)

| 特性 | 含义 | 影响 |
| --- | --- | --- |
| settings | 设置 | 在菜单和仪表板都显示该对象，在设置中能看见 |
| custom | 定制组件 | 会生成默认的定制仪表板和表格 |
| blockchain | 区块链支持 | 自动上链，目前还不支持 |
| iot | 物联网支持 | 物联网节点，目前暂不支持 |