

**用 户 手 册**

**题 目 企业智能数据管理系统的设计和实现--前端**

**模块**

**英文题目Design and realization of enterprise intelligent**

**platform management system -- front-end module**

学生姓名： 毛慧君 申请学位门类： 工学

学 号： 201420181014

专 业： 软件工程

学 院： 软件学院

指导教师： 邓薇 职称： 讲师

**二 0 一 八 年 五 月 三 十 日**

目 录

[1．现实背景 1](#_Toc485415700)

[2．系统技术 2](#_Toc485415701)

[3．功能模块 3](#_Toc485415702)

[4．操作流程图 4](#_Toc485415703)

[5．操作指南 5](#_Toc485415704)

[5.1 系统首页 5](#_Toc485415705)

[5.2 管理员登录 5](#_Toc485415706)

[5.3 管理员管理 6](#_Toc485415707)

[5.4 会员管理 16](#_Toc485415708)

[5.5 游客留言 18](#_Toc485415709)

# 1．现实背景

随着科技的发展，社会的进步，网络信息化时代到来，对数据存储、管理、分析的需求大大提高。我们现在处于信息和数据都极快增长的“大数据”时代，数据是一个企业的核心资产，对于企业发展管理产生深远影响。大数据能增强企业防范风险的能力，使得企业管理决策权成功转变，由被动追求价值向主动增加企业价值转变。再者，企业创造价值的方式也会发生改变，管理决策从单一的中高层管理向员工参与决策转变。因而，企业智能平台管理系统的设计实现势在必行。

在各种类型的不同数据中，大数据技术即快速获得有价值信息的能力。明白这一点十分重要，使得技术具备走向众多企业的潜力也很大程度上是有这一点决定的。大数据的4个特点有四个层面：第一点，数据体量巨大。从TB级别，一直跃升到PB级别；第二，数据类型众多。例如网络、图片、地理位置、日志、视频信息等等。第三，价值密度低。比如说视频，连续不间断监控过程中，也许有用的数据仅仅有一两秒。第四，处理速度极快。现今，信息时代正渐渐向数据时代转变，数据开始变得愈发重要，它能够反映出公司的经营情况，为公司的经营和决策提供准确的参考依据。而数据平台是连接数据和企业的媒介，可以将这些数据通过一定的规格，清晰地展现给领导。

当前的大数据发展非常复杂。尤其是当用户需要挑选适合自身的大数据技术时，往往是踌躇忐忑的，既要考虑这个技术是不是真正解决了自身的问题，又要考虑该技术是否得到了积极快速的发展，还要忧心是不是还有新的技术加入很多能够提升的新功能。事实上，当不同的技术之间差别不是很大的时候，用户是可以随便选某一种技术以所用。但是纵观当今大数据产业的发展，在未来，很大机会是可能出现更加集中化、一体化的技术。

# 2．系统技术

按照系统的生命周期介绍企业智能平台管理，从设计到实现的全过程。该系统前端使用vue框架，后端为java spring技术，再到更底层涉及到信息碎片化以及深度学习和人工智能算法方面为python技术。该系统包括模型管理、碎片化平台、微服务管理、流程管理、权限管理、数据安全管理等部分。按照权限可划分为用户权限、主题权限、密级权限、角色权限、组权限等，可以在其中配置不同用户的权限身份。在本项目中我主要负责前端开发。

前端部分是使用vue-cli工具搭建vue项目框架进行MVVM模块化开发，异步请求接口调用使用axios，它是基于ES6 Promise技术的封装，再者使用vue-router配置项目路由，vuex便于全局变量的管理，其中搭载了webpack打包机制，轻松将代码打包为js、css文件夹和一个index.html入口文件上线。前端的UI框架使用了iView UI，其样式美观，使用便捷，且自定义配置方便。以下为技术详情

3.3.1 Vue.js技术

Vue.js是一套渐进式框架，用以构建用户界面。主要采用自底向上增量开发的设计。Vue的核心库只关注视图层，它不仅容易学习，还便于与第三方库或已有项目。它最大的特点就是双向数据绑定。

主要是先实现一个数据监听器Observer，监听所有数据的对象，如果数据有变化即可拿到新值并通知订阅者，再者实现一个指令解析器Compile，来扫描和解析每个元素的指令，根据指令模板替换数据，以及绑定相应的更新函数。而实现一个Watcher，是连接Observer和Compile的桥梁，能够收到并订阅每个属性变化的通知，执行指令绑定的相应回调函数，从而更新视图。最后mvvm入口函数，整合以上三者。

（1）vue-cli ：vue-cli是Vue.js官方出的脚手架命令行工具，我们可以用它快速搭建好一个Vue.js项目，它主要用来初始化项目，可以使用官方模板，也可以使用自定义模板生成新项目，而且在新版本中，vue-cli新增了build命令，实现了零配置启动Vue.js应用。

（2）vue-router：vue-router是基于Vue.js的路由配置，通过vue-router可以轻松实现项目路由的各种复杂配置。包括可以设置元数据、钩子函数等等。也避免了之前会出现的回调地狱的问题。

（3）vuex：Vuex 是一个专为 Vue.js 应用程序开发的状态管理模式。它采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化。Vuex 也集成到 Vue 的官方调试工具 devtools extension，提供了诸如零配置的 time-travel 调试、状态快照导入导出等高级调试功能。

3.3.2 axios技术

Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中。使用ES6新的promise技术实现异步数据请求，调用后端接口，代码简洁优雅易维护，和Vue.js配套使用，效果更佳。有成熟的文档，方便开发。

3.3.3 webpack技术

本质上说，webpack 是一个现代 JavaScript 程序的静态模块打包器。当 webpack 处理应用程序时，它会以递归方式构建一个依赖关系图，其中包含应用程序需要的各个模块，最后将所有这些模块打包成一个或多个 bundle。它支持高度配置，主要包括四个部分：入口(entry)、输出(output)、loader和插件(plugins)。

3.3.4 iView UI组件

iView是一套基于Vue.js的开源组件库，主要用于PC界面的中后台产品。它高质量，功能丰富，有友好的API可以自由灵活的使用空间。UI细致漂亮，文档详细易读，并且主题也支持高度定制。十分灵活，对于开发者也十分友好。

# 3．功能模块

系统主要包括模型管理、微服务管理、流程管理和权限管理等方面。模型管理包括应用、场景、词库和模型训练；其中应用包括应用列表的显示，以及其相关的增删改操作。场景和词库同理。还包括一个模型模块，可以进行数据训练操作，训练好的数据会直接显示在其下方的训练列表中。微服务管理包括微服务的增删改；流程管理包括流程的增加、编辑、删除。

图3-1 系统的功能结构图

# 4．操作流程图

用户先输入用户名和密码，验证登录成功后即可进入应用列表首页，在首页可以选择某一项直接跳转到该应用下的场景，也可以选择编辑或删除场景；场景列表下也可以编辑和删除操作等，词库也是同理。模型训练页面可以选择需要训练的应用选择训练进行训练，然后刷新页面可看到训练好的列表。而进入流程管理和微服务管理中进行浏览、编辑、添加和删除的操作。下面用业务流程图来说明：

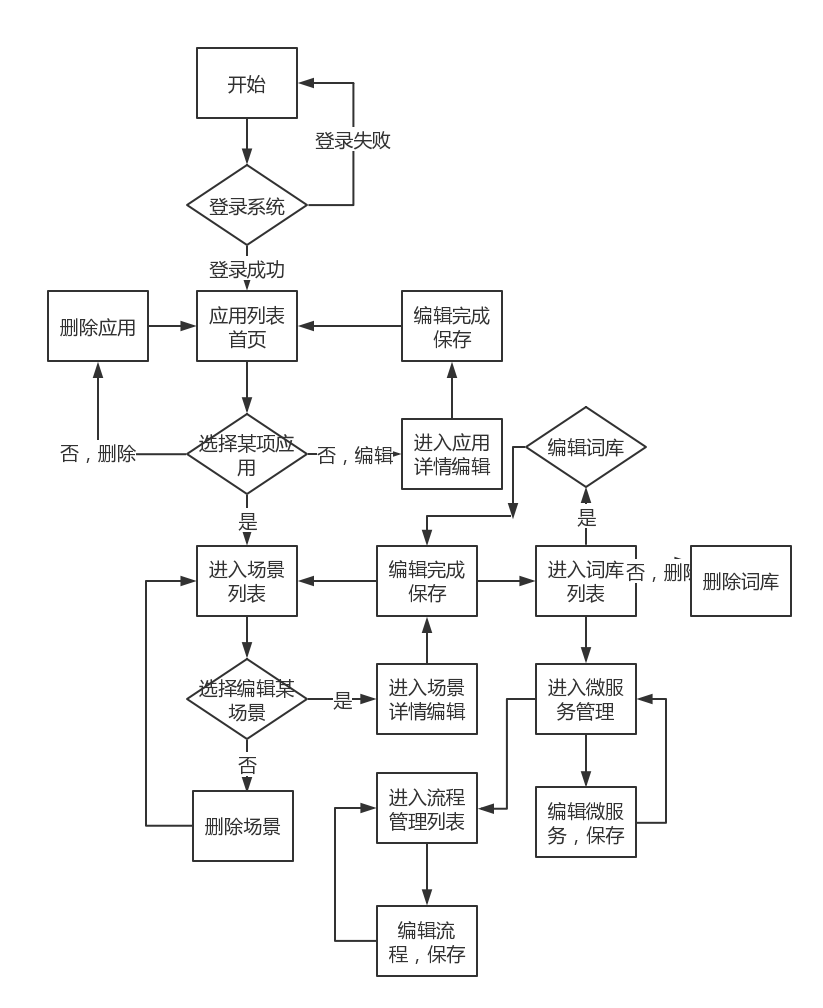


图4-1 操作流程图

# 5．操作指南

5.1 系统设置

系统设置方面主要包括用户的登录、注册以及修改密码和退出等。

5.1.1 登录和注册

根据表单的提示，在登录前需要注册用户信息，输入用户名并设置密码，点击登录显示注册成功信息即可。需要登录时输入之前注册过的信息，如果用户名和密码确实存在且匹配则直接进入首页，表示登录成功。

页面截图：



图5-1 登录/注册页面

5.1.2 登出

在页面的右上方，有一个设置选项旁边有一个向上的箭头，将鼠标悬浮在上面，小箭头向下，显示出‘安全退出’的下拉框。点击‘安全退出’，页面返回到登录页，退出成功。

页面截图：

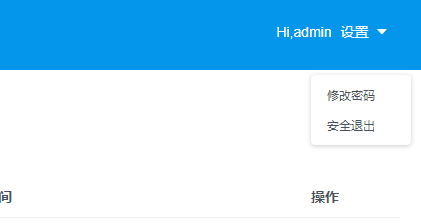


图5-2 登出

5.1.3 修改密码

同样，在页面的右上方位置的设置，点击后出现修改密码。点击修改密码，出现修改密码页面，按照页面提示，先输入登录原密码，再输入希望修改后的密码，点击确定跳转到登录页，使用刚才修改的新密码登录即可。如果在修改密码时原密码输入错误，则不能进行修改密码操作。

页面截图：

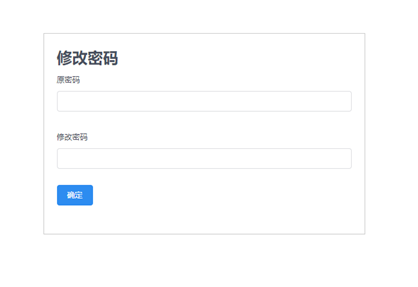


图5-3 修改密码

5.2 模型管理

模型管理是该系统中的重要部分，主要是进行模型数据的训练和配置。尤其在场景部分，数据和用户交互比较复杂。

5.2.1 应用列表

应用列表主要是对用户新建的应用在表格中进行展示，同时也便于用户对其需要的部分进行相关编辑操作等。默认登录后直接跳转到这一页。当然，点击左侧应用，也可跳转到这一页。在应用列表中，右侧有编辑和删除操作，点击编辑跳转到该应用的详情页，点击删除会弹出模态框，需要你确认删除。如果直接点击某个应用那行，会跳转到该应用下的场景列表页面。再者，页面上方的时间和关键词可以对应用进行搜索。

页面截图：

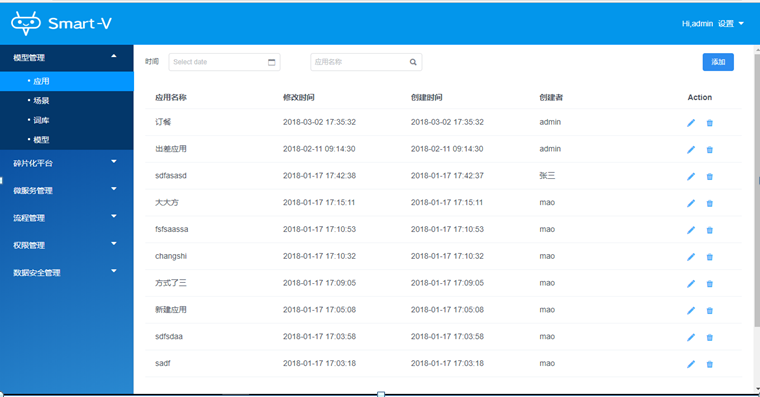


图5-4 应用列表

5.2.2 添加、编辑应用

添加编辑主要是实现对应用的相关操作，可以对既有的应用进行修改也可以添加新的应用。进入选择编辑进入编辑页面，可以做想要的修改，在选择保存即可。而添加也是同理。

页面截图：

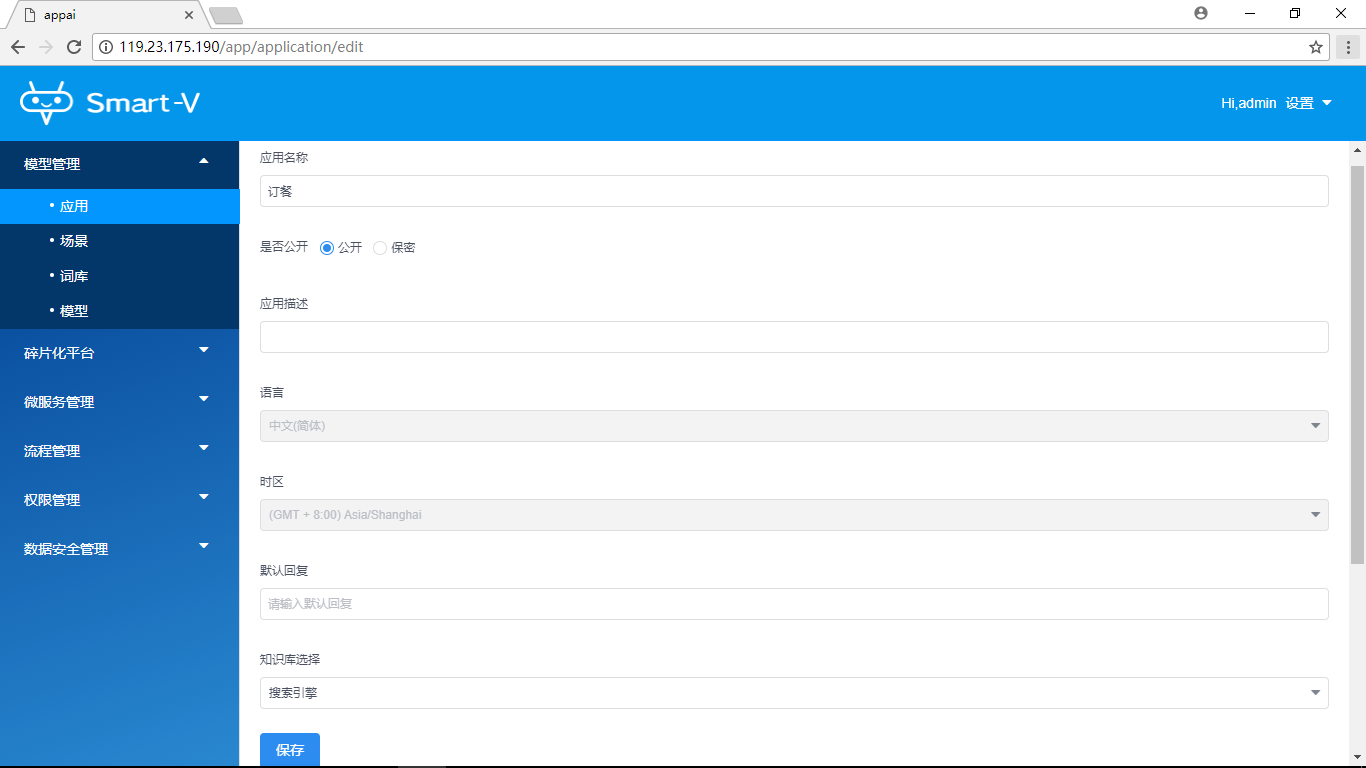


图5-5 应用编辑

5.2.3 删除应用

删除某应用，可以在编辑应用详情中选择删除，也可以在应用列表中点击删除图标进行快速删除。并回到应用列表或刷新列表。选择删除时会弹出模态框，用户再次确认删除后即可删除应用。

页面截图：

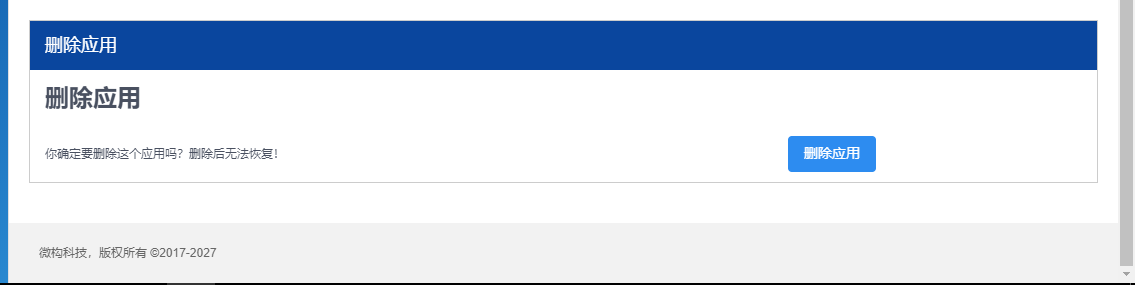


图5-6 删除应用

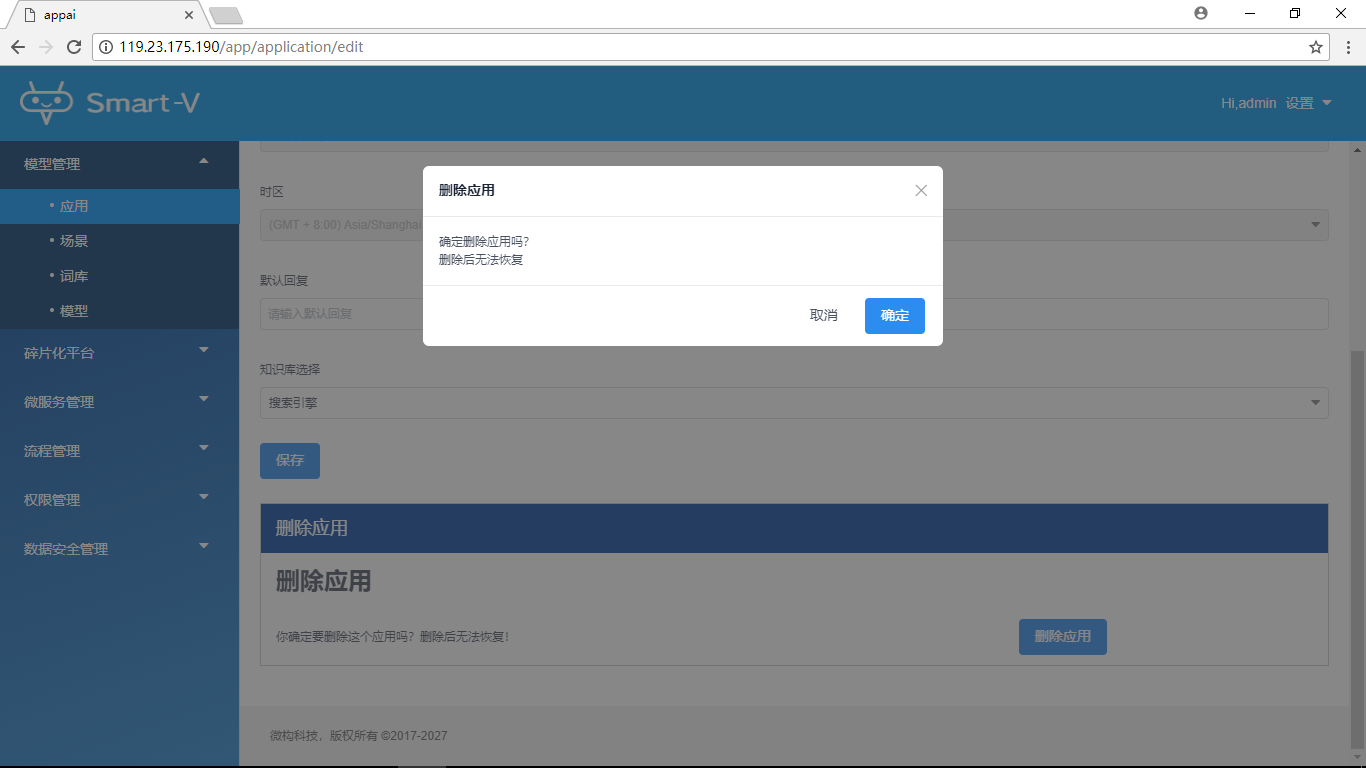


图 5-7 删除应用

5.2.4 场景列表

场景一定是归属于应用，如果没有选择应用直接进入场景列表则会提示需要先选择应用，再跳转回应用列表让用户选择。如果已经选择了某个应用会直接跳转到这个应用下的场景列表，在场景列表中可以直接点击列表右侧图标进行编辑和删除操作，页面上测可以进行相关内容筛选，比如时间或关键字等。

页面截图：

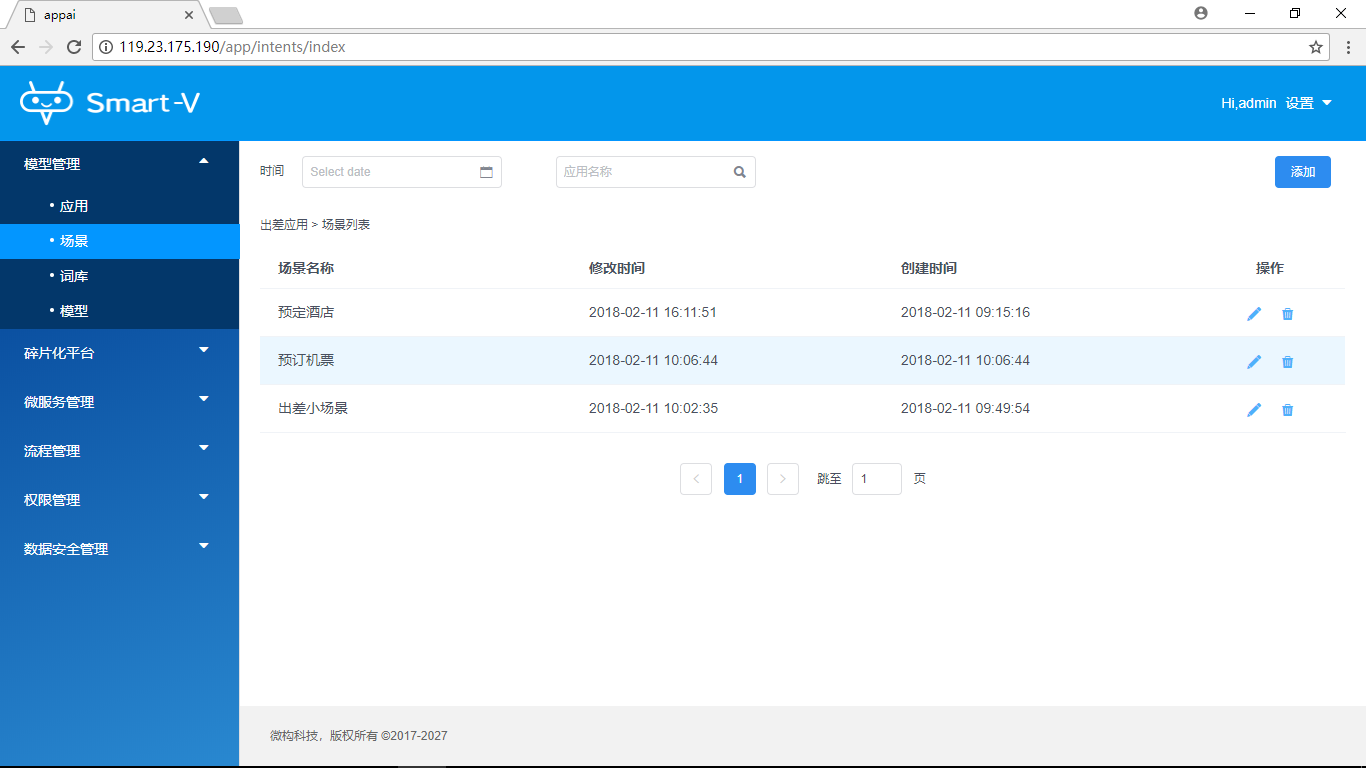


图 5-8 场景列表

5.2.5 添加、编辑场景

在场景列表页面点击编辑，跳转到场景的详情页面，用户可以针对需要修改的部分重新对该场景详情进行调整，点击保存即可。而添加也是跳转到详情页，不过该页面为空白，所有内容都要开始添加。

页面截图：

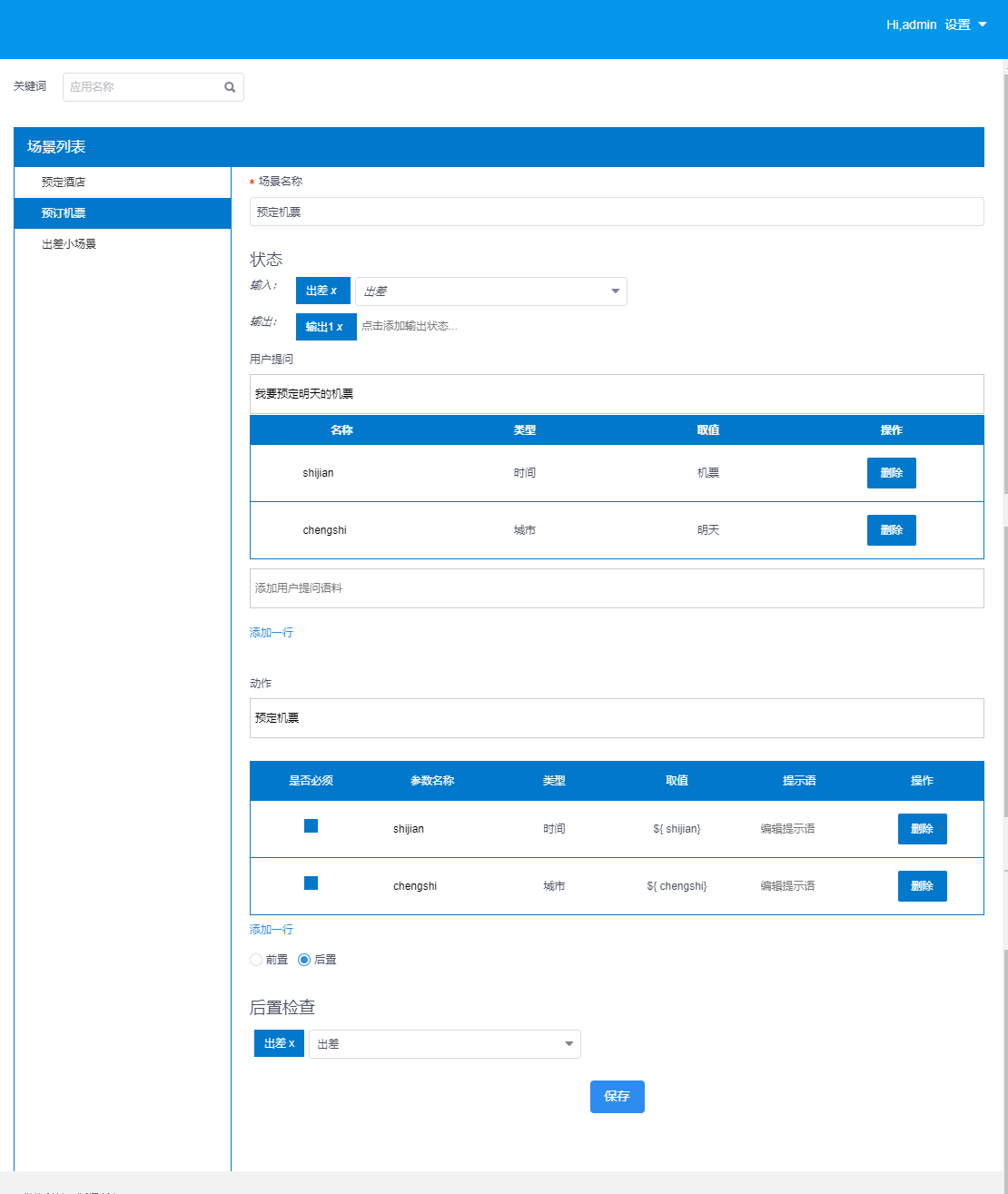


图 5-9 添加、编辑场景

5.2.6 删除场景

在场景列表处点击删除按钮，会跳出弹框询问用户是否确定要删除这个场景，如删除后则不可恢复，如果用户确认要删除，即可删除该场景。删除成功后会重新刷新场景列表页，显示最新的场景列表。

页面截图：

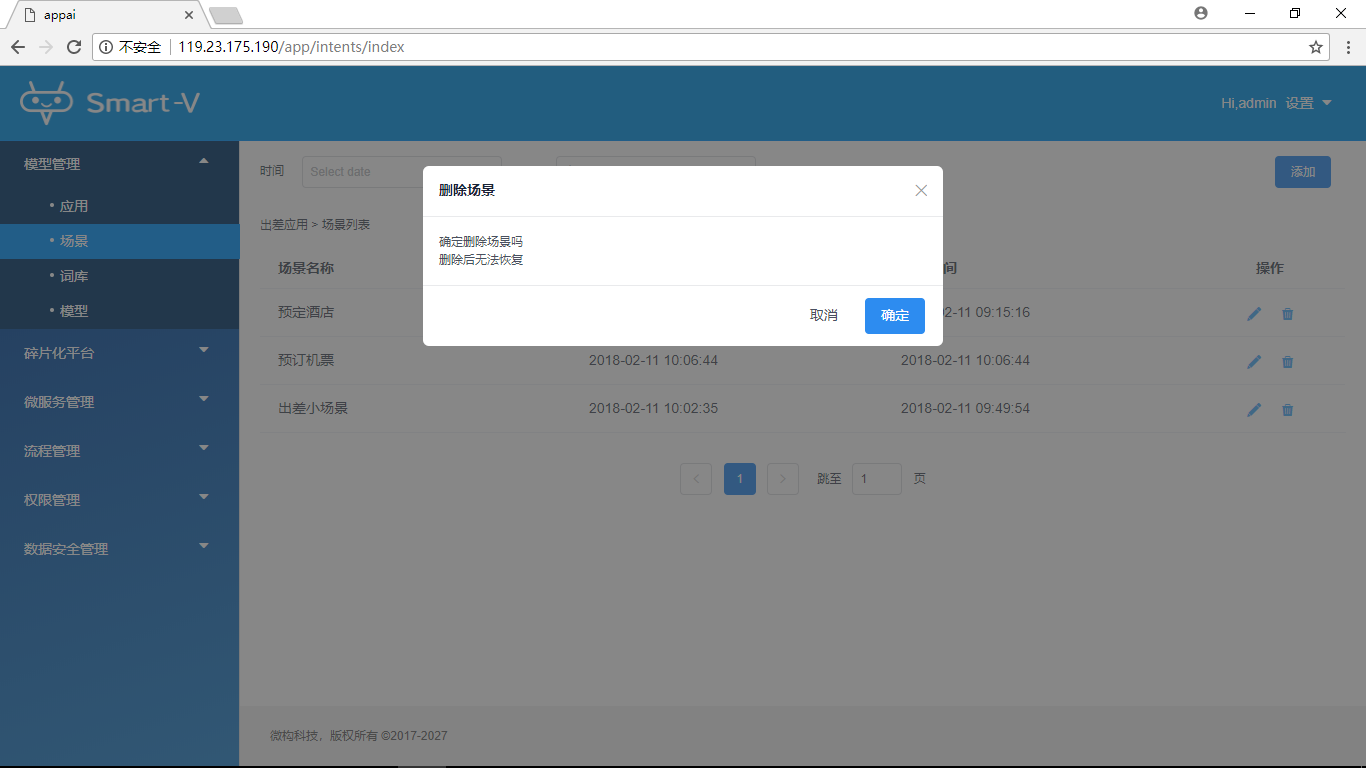


图 5-10 删除场景

5.2.7 词库列表

可以在左侧的蓝色菜单列表中选择词库，即可跳转到词库列表，与场景列表一样，词库列表一定归属于某个应用，如果没有选择应用，不能直接进入词库列表，会提示错误信息并直接跳回应用列表要求用户必须选择一个应用。在词库列表中，左侧操作部分可以进行编辑和删除工作，上方有日期和关键字的搜索框等。

页面截图：

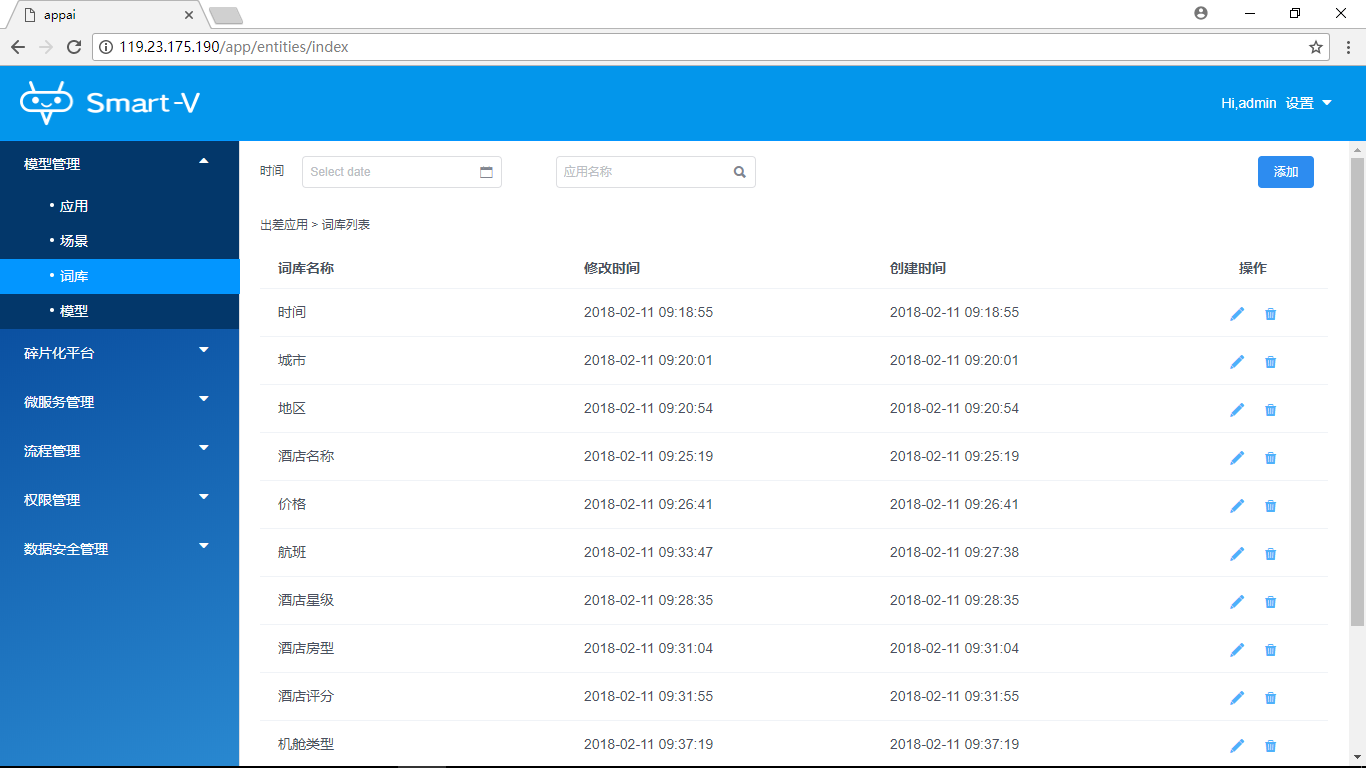


图 5-11 词库列表

5.2.8 添加、编辑词库

在词库列表页面，点击编辑按钮即可跳转到词库的详情页

页面截图：

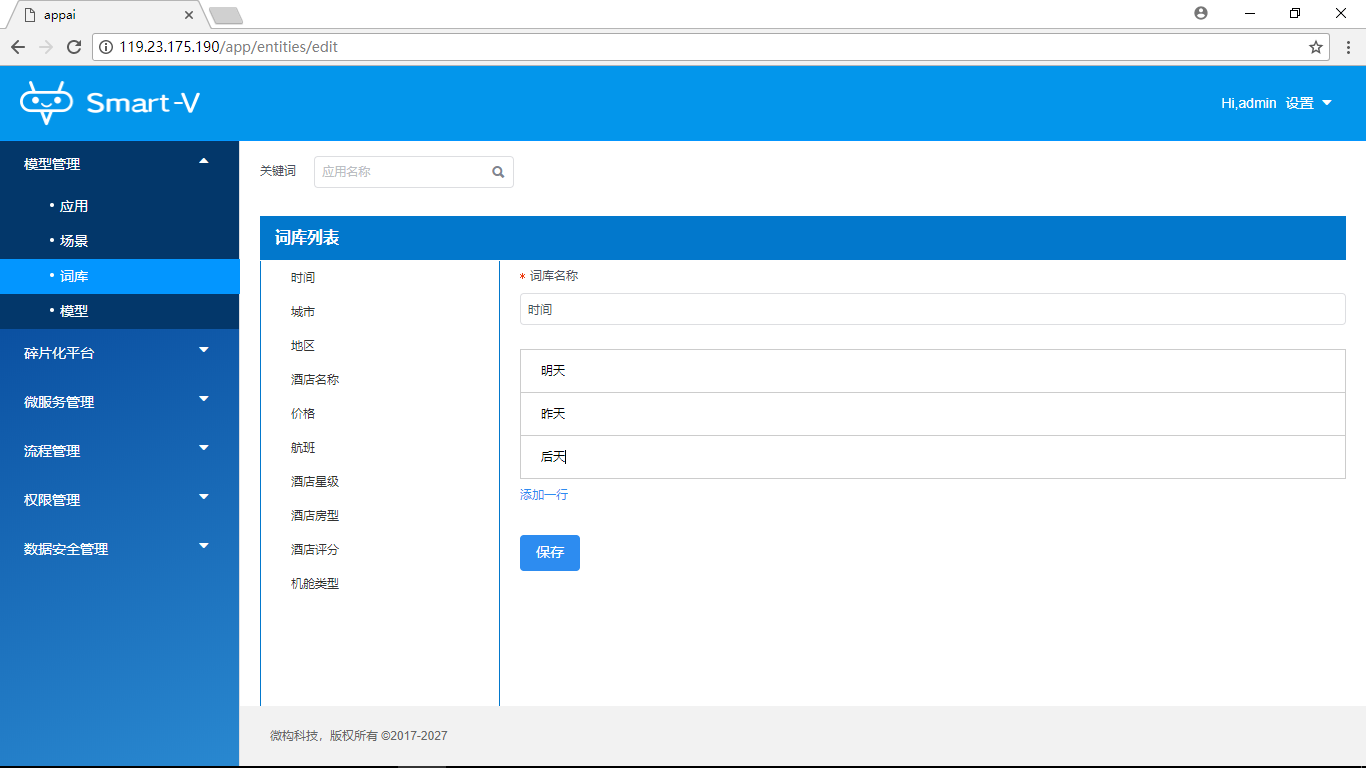


图 5-12 添加、修改词库

5.2.9 删除词库

在词库列表中点击删除按钮，即可将要删除的词库id保存到全局中，点击确定时，就调用删除接口，并将词库id传递给删除接口。即封装delEntities()函数，调用后台‘dict/del’接口，如果返回的response.data为空，则表示操作成功，会提示用户删除成功，再调用getEntitiesList()，重新刷新词库列表。

页面截图：

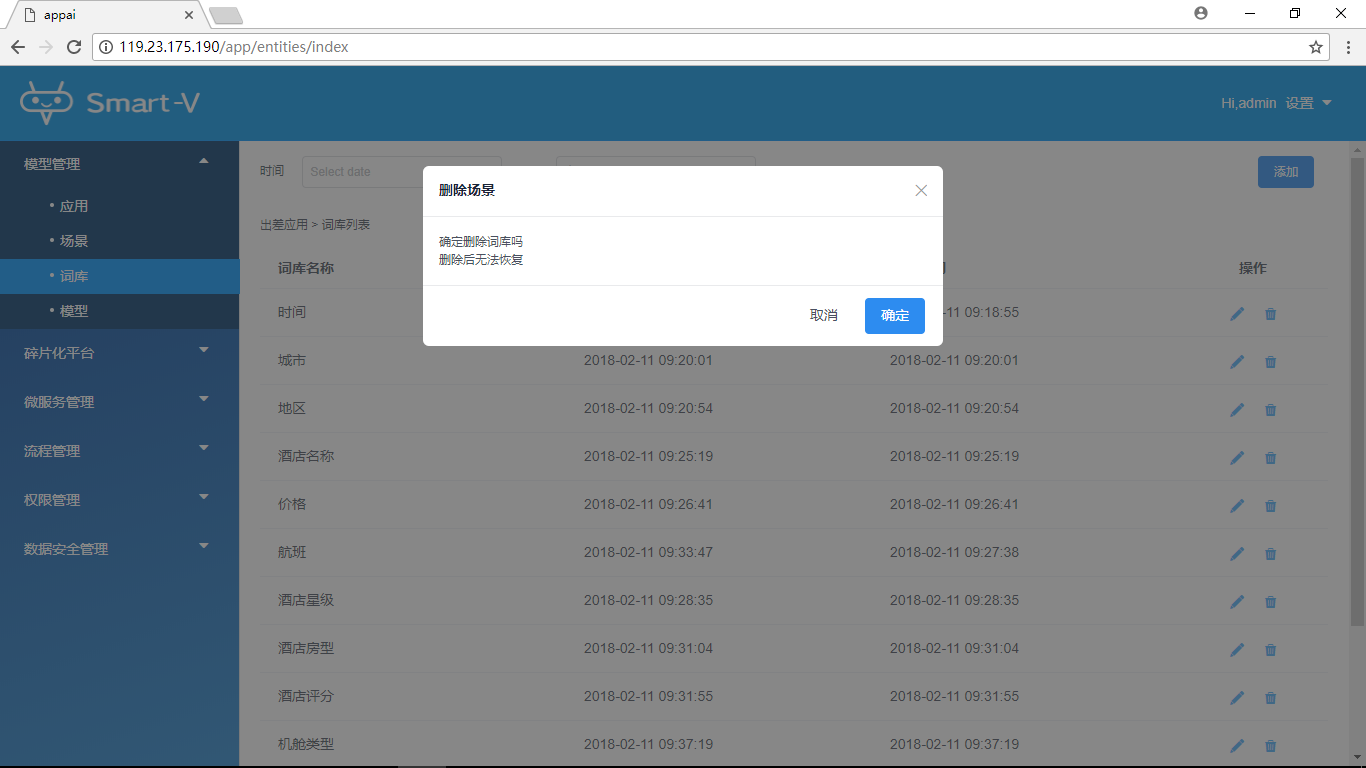


图 5-13 删除词库

5.2.10 模型训练

下拉框中是所有的应用名称选项，可以选择某一项应用进行数据训练，这边主要是对接了python的接口，进行操作。下拉框列表显示所有应用，因此首先需要调用getAppList()，返回所有应用。选中某一项后确定训练，将会生成训练数据，调用doTrain()函数，它需要获取选中的应用id，若返回data为null，则提示数据训练成功。再调用getTransList()显示所有训练好的数据列在下方表格中。已经训练过的应用数据不可重复训练。

页面截图：

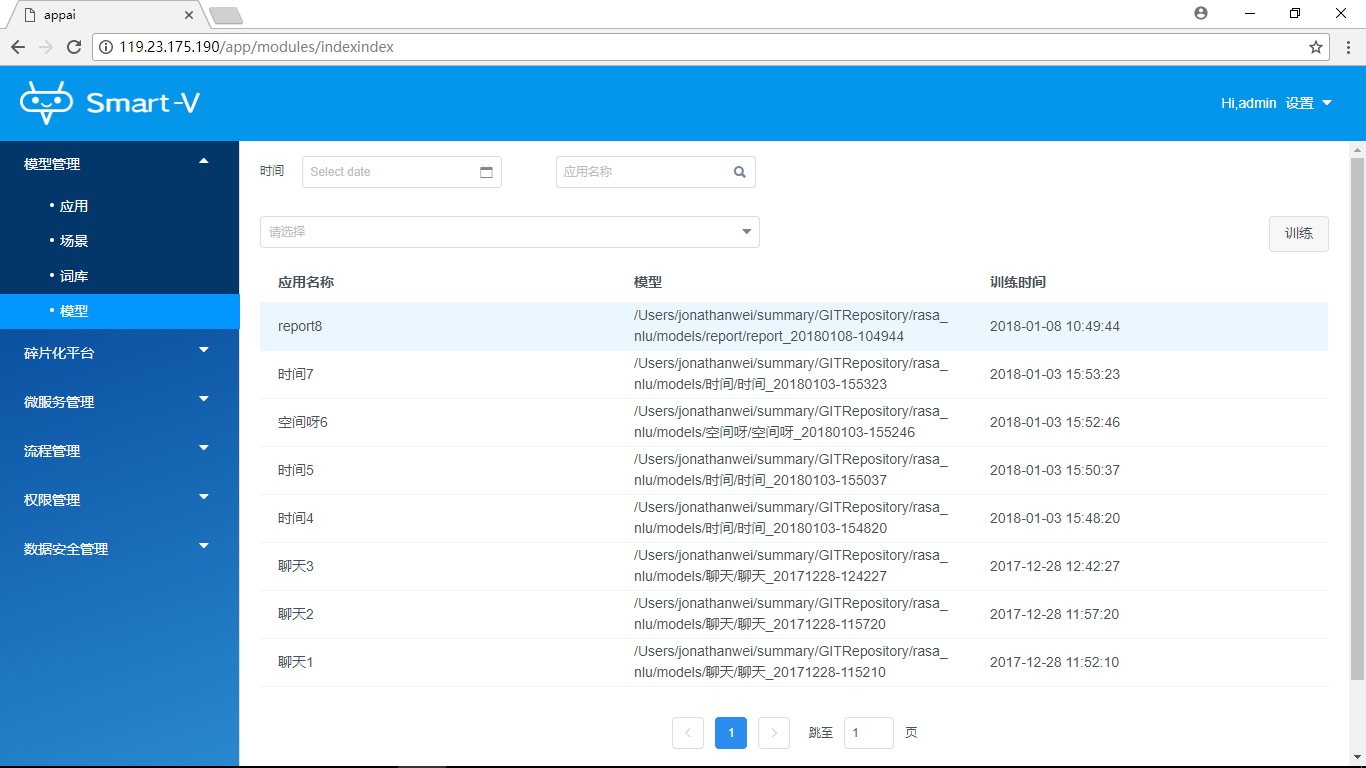


图 5-14 模型训练

5.3 微服务管理

5.3.1 微服务管理列表

微服务管理列表列出了所有应用下的微服务，以及它对应的描述、接口、url等等。用户可以随时对其进行删除、编辑等操作。选中某项微服务将microId 存入sessionStorage保证页面在刷新时，仍能保存相关数据。同时，该列表也支持关键词和日期模糊查询。封装函数getMicroList()，获取相关数据微服务名称关键词（name）、日期（date）、每页信息数（pageSize）、当前页码（pageNo），再调用后台接口‘mic\_service/list’返回微服务信息列表。在页面刷新时调用getMicroList()。

页面截图：

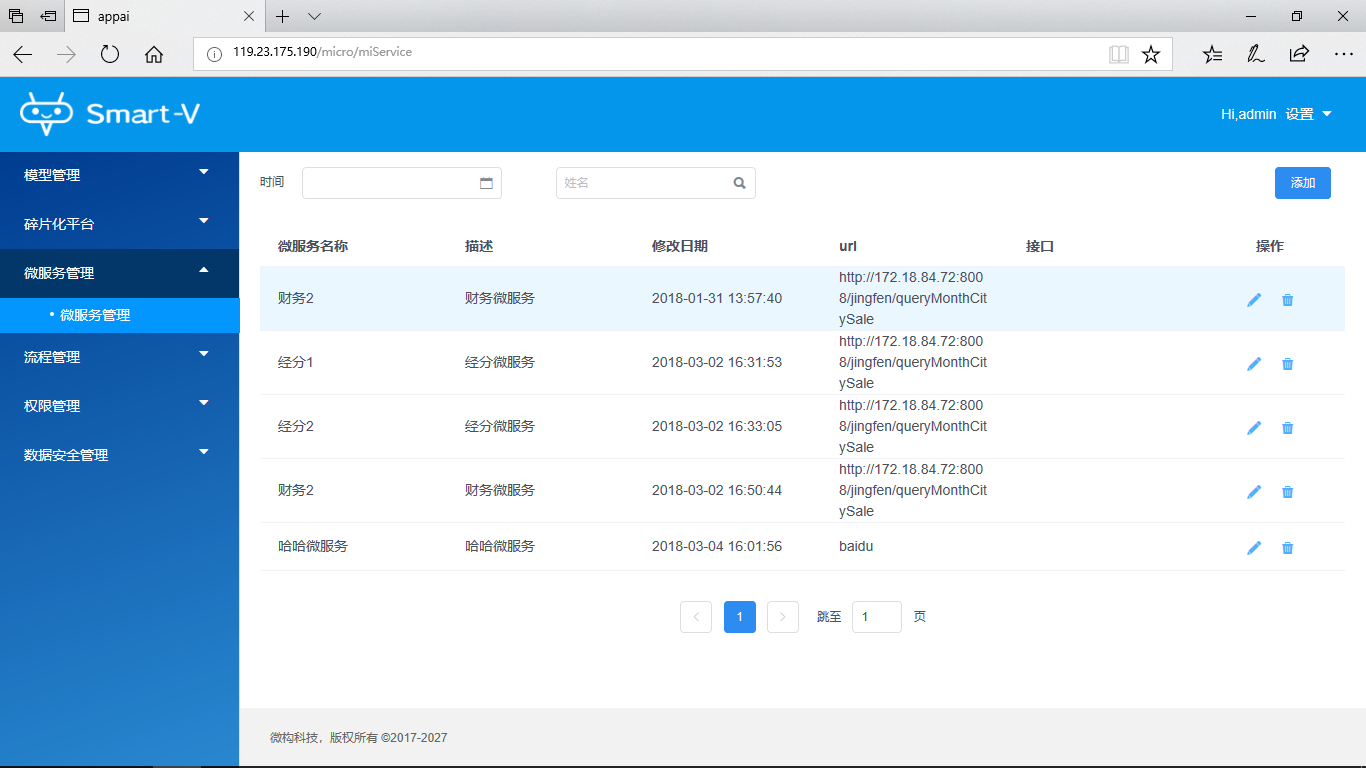


图 5-15 微服务管理

5.3.2 添加、编辑微服务

一个微服务中可以存在多个参数和值，而名称描述和url只有一个，因此在添加、编辑页面中可以另外多添加几个参数，点击添加时弹出模态框，对需要内容进行填充，最后保存所有修改内容即可。编辑时首先查询当前微服务详情信息显示在页面上，即调用getMicriDetail()，它需要获取当前需要编辑微服务的id（microId），之后即可返回微服务详情。编辑完成后选择保存，即调用saveCreate()，将新的数值提交给后台，注意添加时，将microId的值置为空即可。

页面截图：

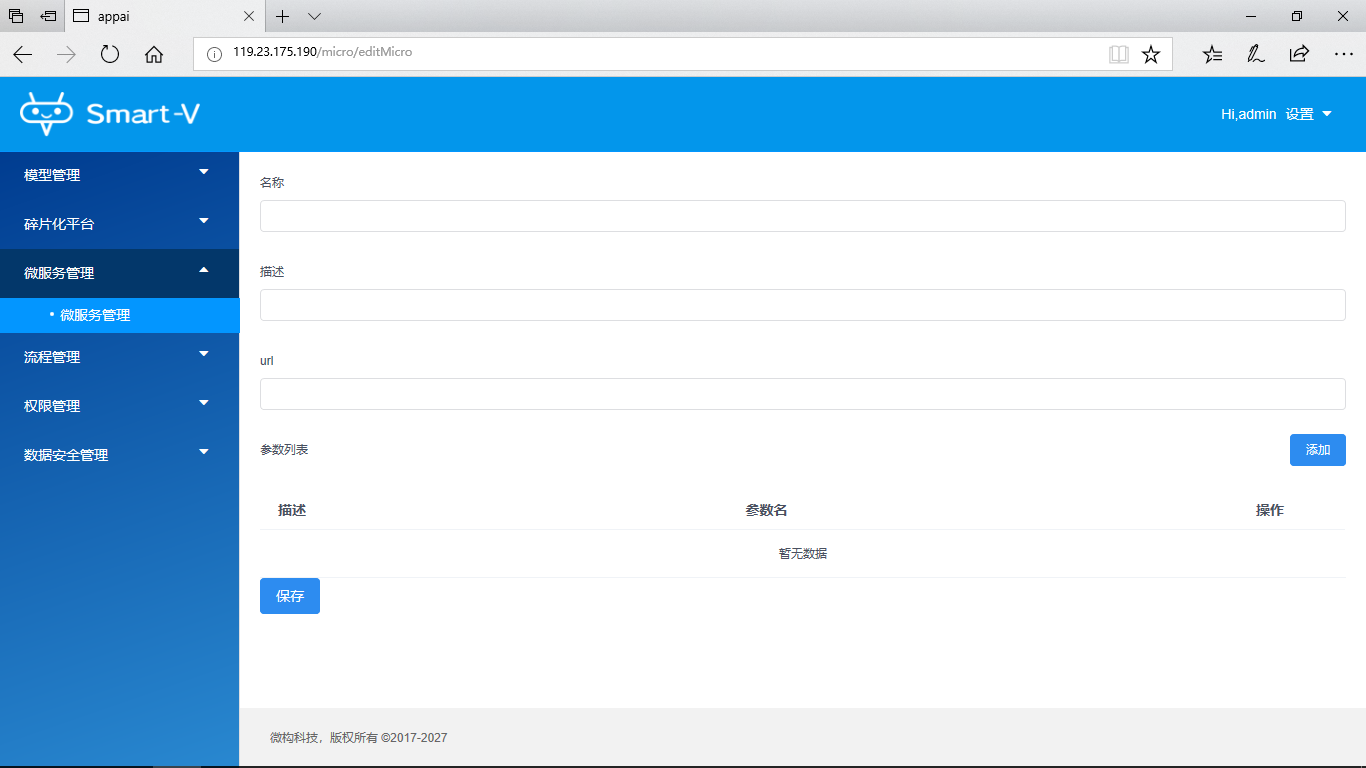


图 5-16 添加、编辑微服务

5.3.3 删除微服务

在微服务列表中，选中需要删除的某一项弹出模态框，请用户确认是否要删除，点击确定按钮，删除该项微服务，隐藏模态框并刷新微服务列表。封装函数deleteMicro()，获取需要删除的微服务id（delId），将它传给后台调用‘mic\_service/delete’，如果response.data 为null，就提示删除成功，否则表明删除失败。

页面截图：

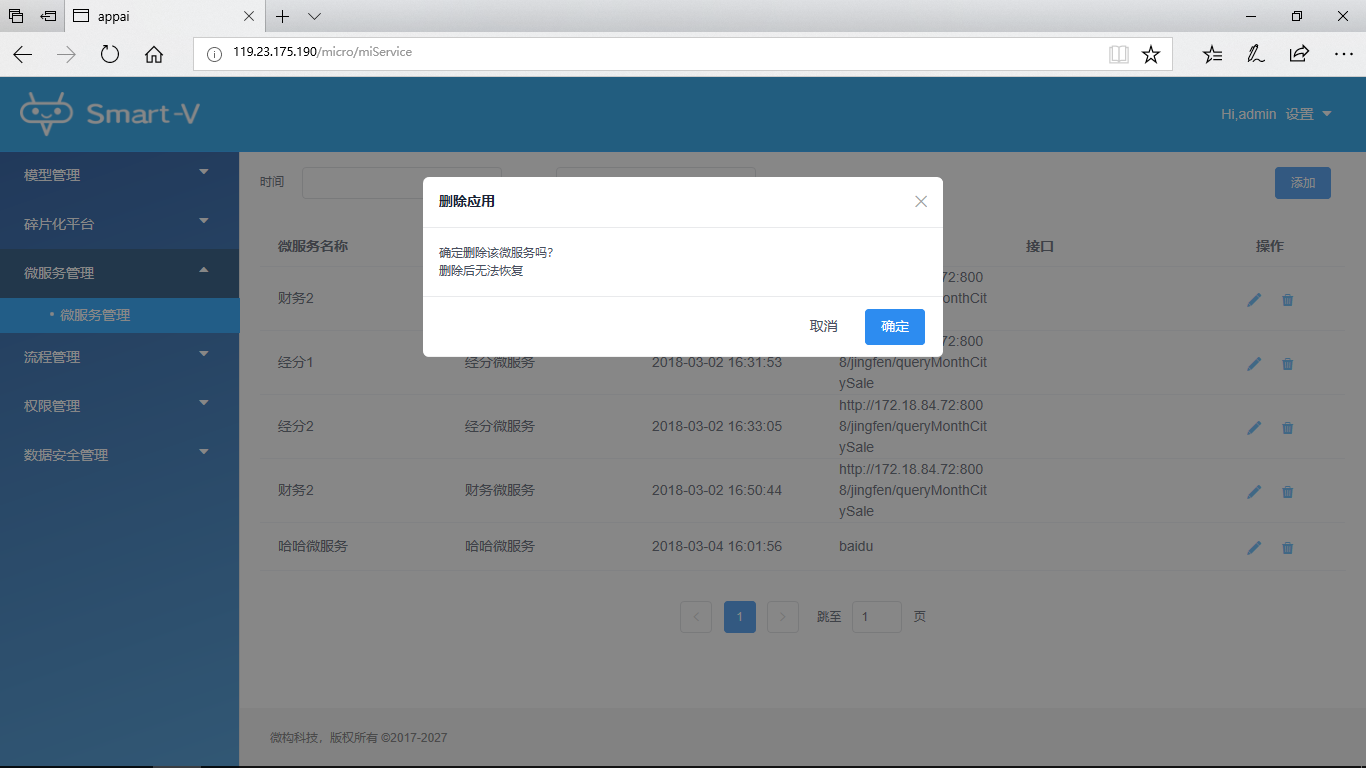


图 5-17 删除微服务

5.4 流程管理

流程管理包括流程管理列表显示、编辑和添加流程管理、删除流程管理等。

5.4.1 流程管理列表

流程管理列表主要展示了流程列表信息，管理了数据流程。能够实现关键字查询和按日期搜索以及翻页的功能。主要通过封装getFlowList()函数，获取日期（date）、流程名称关键字（name）、页面信息数量（pageSize）、当前页码（pageNo）等。调用后台接口‘process/list’，如果返回response.data不为空，则将返回数据赋值给数组processList在页面上显示出来。在页面刷新时调用该函数。

页面截图：

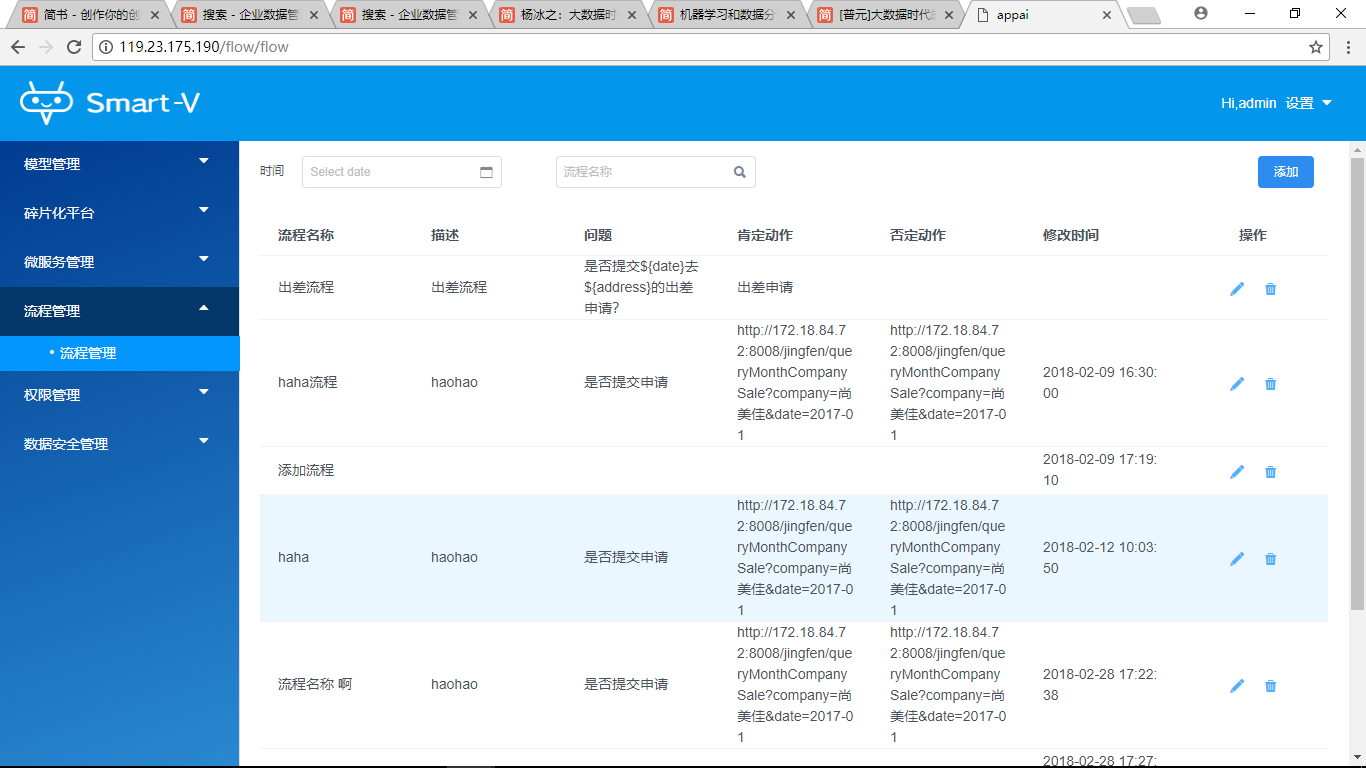


图 5-18 流程管理列表

5.4.2 添加、编辑流程

在流程列表中选择编辑，即可实现重新改写流程详情内容，也可以新添加新的流程，二者极为相似，不同点在于编辑前需要调用一次详情接口并在保存时，再次提交保存流程的id，而添加直接跳转添加页面，并将id赋值为null即可。编辑时首先需要获取需要修改流程的详情信息，即调用getDetail()函数，该函数需要先获取流程id（flowId）,即可返回processBlockList详情信息，显示在页面上。修改后保存saveCreate()即可，显示成功后跳转流程列表页面。

页面截图：

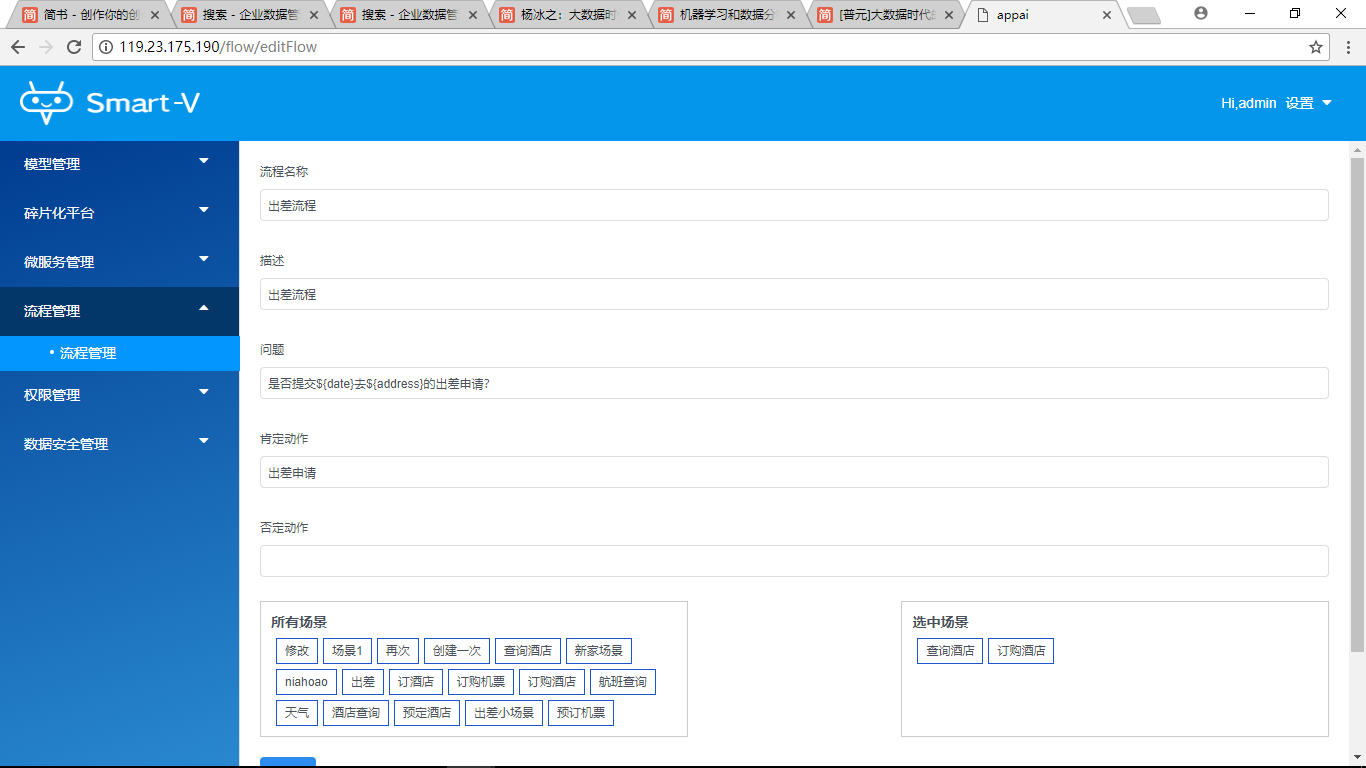


图 5-19 添加、编辑流程

5.4.4 删除流程

在流程列表中点击删除图标，弹出模态框再次询问是否确认删除该流程。再次选择确认后，对该流程进行删除操作，并重新刷新流程列表。封装函数deleteFlow()，获取需要删除的流程id（flowId），如果response.data 的值为空，则提示删除成功，否则提示删除失败。最后自动刷新流程列表，显示最新的流程列表。

页面截图：

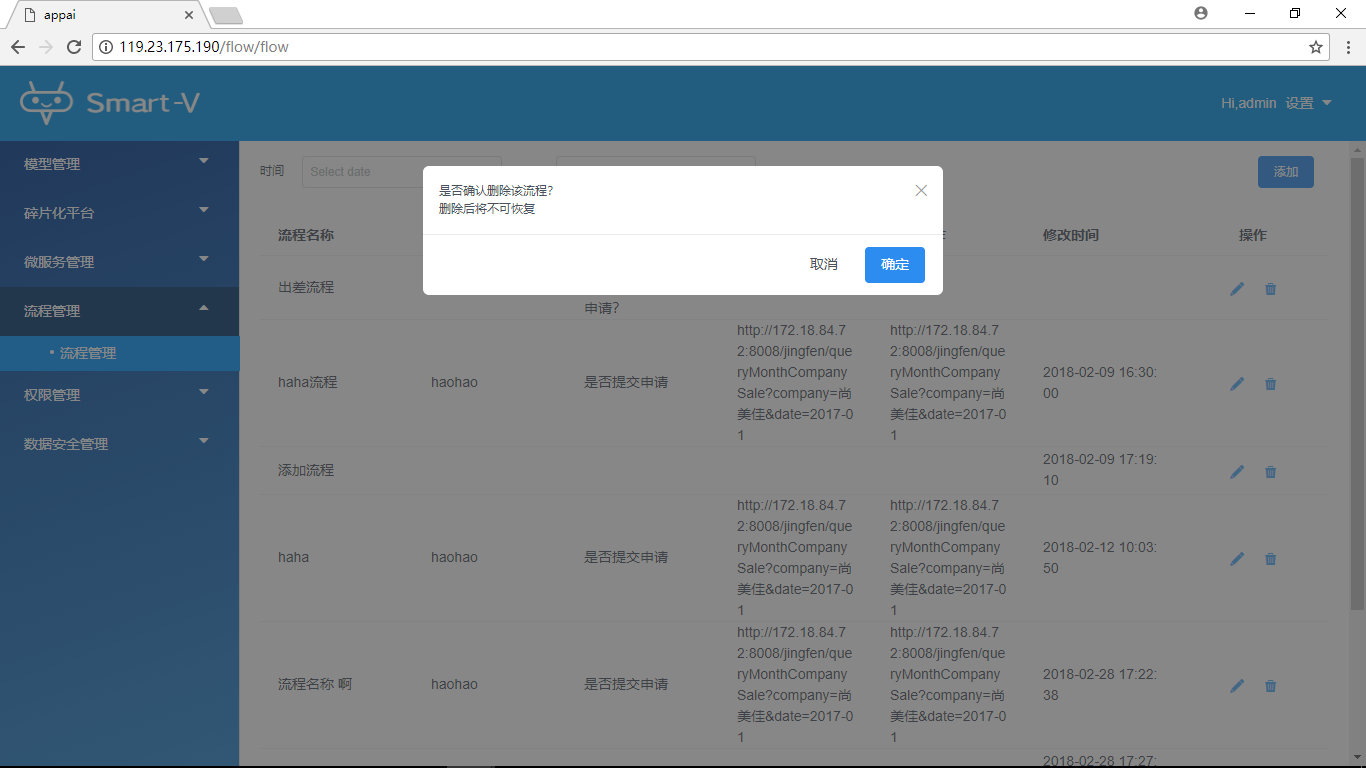


图 5-20 删除流程

感谢广大校园教工人员以及学生用户使用本系统，欢迎各位用户不断提出建议，本人今后会根据各位用户提供的意见对系统进行不断地完善，谢谢！