*南京大学*

*软件学院*

*化学品危险信息流转识别系统*

*设计文档*

目录

目录 2

表目录 3

图目录 4

1、引言 5

1.1 编制目的 5

1.2 词汇表 5

1.3 化学品信息来源 5

2、产品概述 6

3、微信小程序框架概述 6

4、系统总体设计 7

4.1 用户角色和需求 7

4.2 系统总体结构 7

4.3系统架构 8

4.4 系统数据库设计 9

4.4.1 数据库E-R图 9

4.4.2 数据库表 10

4.5 系统包图 11

4.6 系统详细类图 12

4.7 系统时序图 13

4.7.1 扫描二维码查询流程 13

4.7.2 根据名称查询流程 13

4.7.3 根据CAS码查询流程 14

4.7.4 查询历史详情流程 15

4.7.5 报警流程 16

4.7.6 查看报警信息流程 17

表目录

表1.1 词汇表 6

表4.1 用户角色和功能需求表 8

表4.2 化学品基本信息及危害信息表关键属性 12

表4.3 报警信息表关键属性 12

图目录

图4.1 微信小程序框架架构图 7

图4.1 系统总体结构图 9

图4.2 系统架构图 10

图4.3 系统E-R图 11

图4.4 系统包图 12

图4.5 详细类图 13

图4.6 扫描二维码查询时序图 14

图4.7 根据名称查询时序图 15

图4.8 根据CAS码查询时序图 16

图4.9 查询历史详情时序图 17

图4.10 报警时序图 18

图4.11 查看报警信息时序图 19

# 1、引言

## 1.1 编制目的

本文档详细完成对危险化学品危险信息流转识别系统的详细设计，达到指导后续软件构造的目的，同时实现和测试人员及用户的沟通。

本文档面向开发人员、测试人员及最终用户而编写，是了解系统架构的导航。

## 1.2 词汇表

|  |  |
| --- | --- |
| **词汇名称** | **词汇含义** |
| \_view | 视图层 |
| \_app\_service | 逻辑层 |
| \_native | 系统层 |

表1.1 词汇表

## 1.3 化学品信息来源

[1] 中国化工产品大全（第四版），化学工业出版社，北京：2012年；

[2] 中国化工信息中心（CNCIC），中国化工产品及生产厂家数据库，2014年；

[3] ICSC, 国际化学品安全卡: No. 0155, 2005;

[4] European ECHA CHEM, Registration Numbers: 01-2119485561-34-0000, Jan. 16, 2013;

[5] Japan GHS Classification Results+A1:Y87 ID 12, Mar. 23, 2006;

[6] 严格限制进出口有毒化学品信息表，2010年;

[7] 百度百科：No. 77663，2014年。

# 2、产品概述

化学品危险信息流转识别系统以信息编码、自动识别、物联网、数据库、移动互联网、微信小程序等信息技术为基础，借助移动客户端自动扫描或手动输入方式，从而研发可便捷查询化学品危害信息、防范应急措施及相关生产使用活动信息，和上报化学品危险报警信息的一项微信小程序。

# 3、微信小程序框架概述

微信小程序框架包含两部分：View视图层和App Service逻辑层。视图层和逻辑层在两个线程里运行。视图层通过WebView来渲染页面，而逻辑层通过JSCore来请求数据，调用接口等。视图层和逻辑层通过系统层的JSBridage进行通信，逻辑层把数据更新通知道视图层，触发视图层页面更新，视图层把触发的事件通知到逻辑层进行逻辑处理。微信小程序的框架架构图如图4.1。

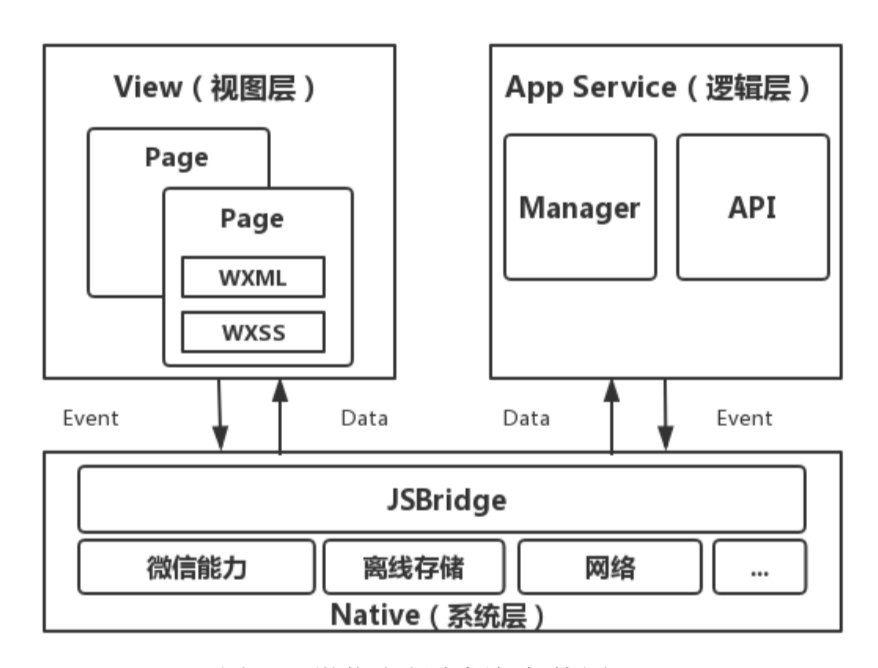


图4.1 微信小程序框架架构图

（1）微信小程序的结构。小程序由描述整个程序的app和描述多个页面的page组成。负责小程序全局配置包含app.js、app.json、app.wxss三个文件。每个page则包含相应页面的js、wxml、json、wxss四个文件。

（2）配置。由全局的app.json或者每个page的json文件进行配置。

（3）逻辑层。逻辑层使用JS进行开发。逻辑层接受视图层的事件响应，同时，将数据处理后传给视图层。在此基础上，增加App和Page方法，对app或者是page进行注册；提供微信支付、扫一扫等基础功能；增加getApp和getCurrentPages方法，获取App的实例和当前页面的实例；每个page的作用域独立。

（4）视图层。页面结构由WXML描述，页面的样式由WXSS设置，组件展示。视图反映从逻辑层获取到的数据，同时将视图层的监听到事件传给逻辑层。 （5）系统层。主要包含网络、存储、文件系统和多线程等。

# 4、系统总体设计

## 4.1 用户角色和需求

根据化学品危险信息流转识别系统的需求，本项目主要的用户分为两种，即普通用户和管理员，用户角色和其功能需求如表5.1所示。

|  |  |
| --- | --- |
| **用户** | **行为** |
| 普通用户 | 扫描二维码查询化学品信息  输入名称查询化学品信息  输入CAS号查询化学品信息  查看搜索历史记录  查看化学品信息详情  上传流转信息  上传报警信息 |
| 管理员 | 查看报警信息 |

表4.1 用户角色和功能需求表

## 4.2 系统总体结构

系统从模块上看主要由普通用户端和管理员端组成，其中普通用户端分为查询模块、查看化学品详情模块、上传信息模块；管理者端主要为查看报警信息模块。系统的所有模块都是web端。

根据系统角色和功能需求，化学品危险信息流转识别系统的总体结构图如图4.1。

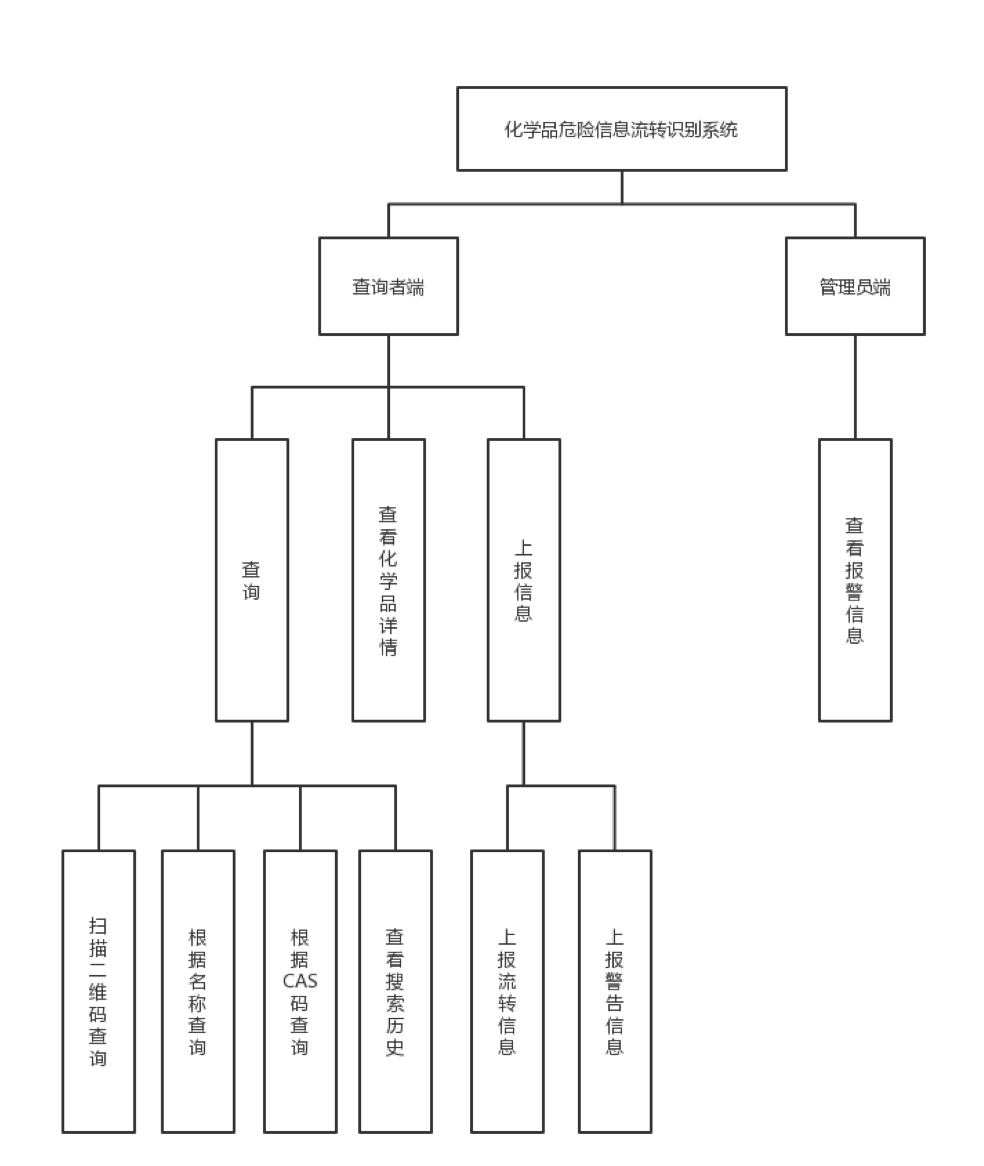


图4.1 系统总体结构图

## 4.3系统架构

以用户端架构为例，阐述化学品危险信息流转识别系统的架构。系统采用前后端分离的开发方法。前端负责人机交互和数据展示，后端负责处理请求。系统架构图如图4.2。

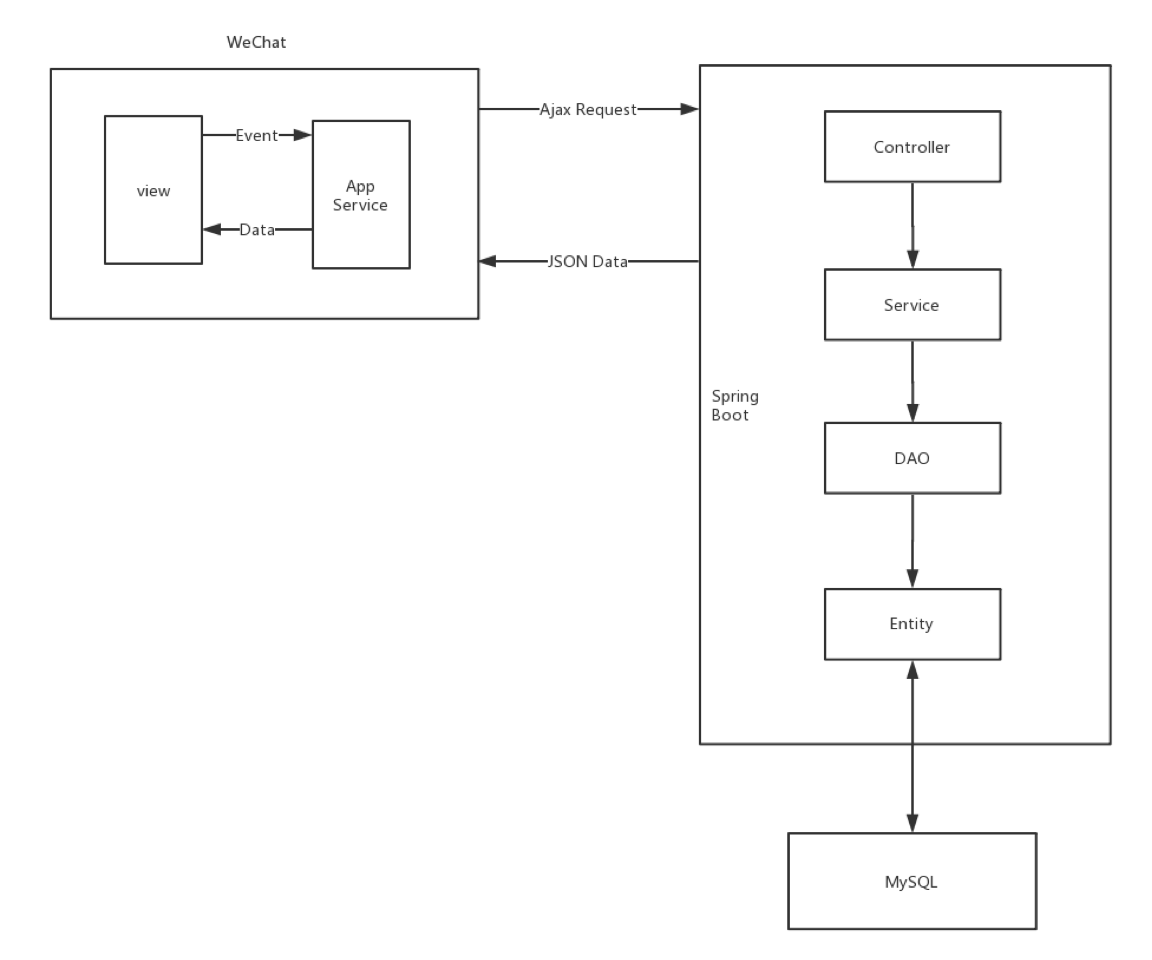


图4.2 系统架构图

如图4.2可知，当小程序端触发了事件，JS负责向后端发送请求，后端的Controller层处理请求，控制业务模块流程，调用Service层的接口来控制业务流程，Service层主要负责业务模块的应用逻辑，具体要调用已经定义的DAO层接口，DAO层负责实现对象和数据库的映射。对于每张表来说，都有一个对应的DAO类，DAO类对根据业务完成相应的增删改查操作。后端对请求处理完成后，返回给小程序JSON数据，在前端展示结果。

## 4.4 系统数据库设计

### 4.4.1 数据库E-R图

基于微信小程序的化学品危险信息流转识别系统的E-R图如图5.3所示。

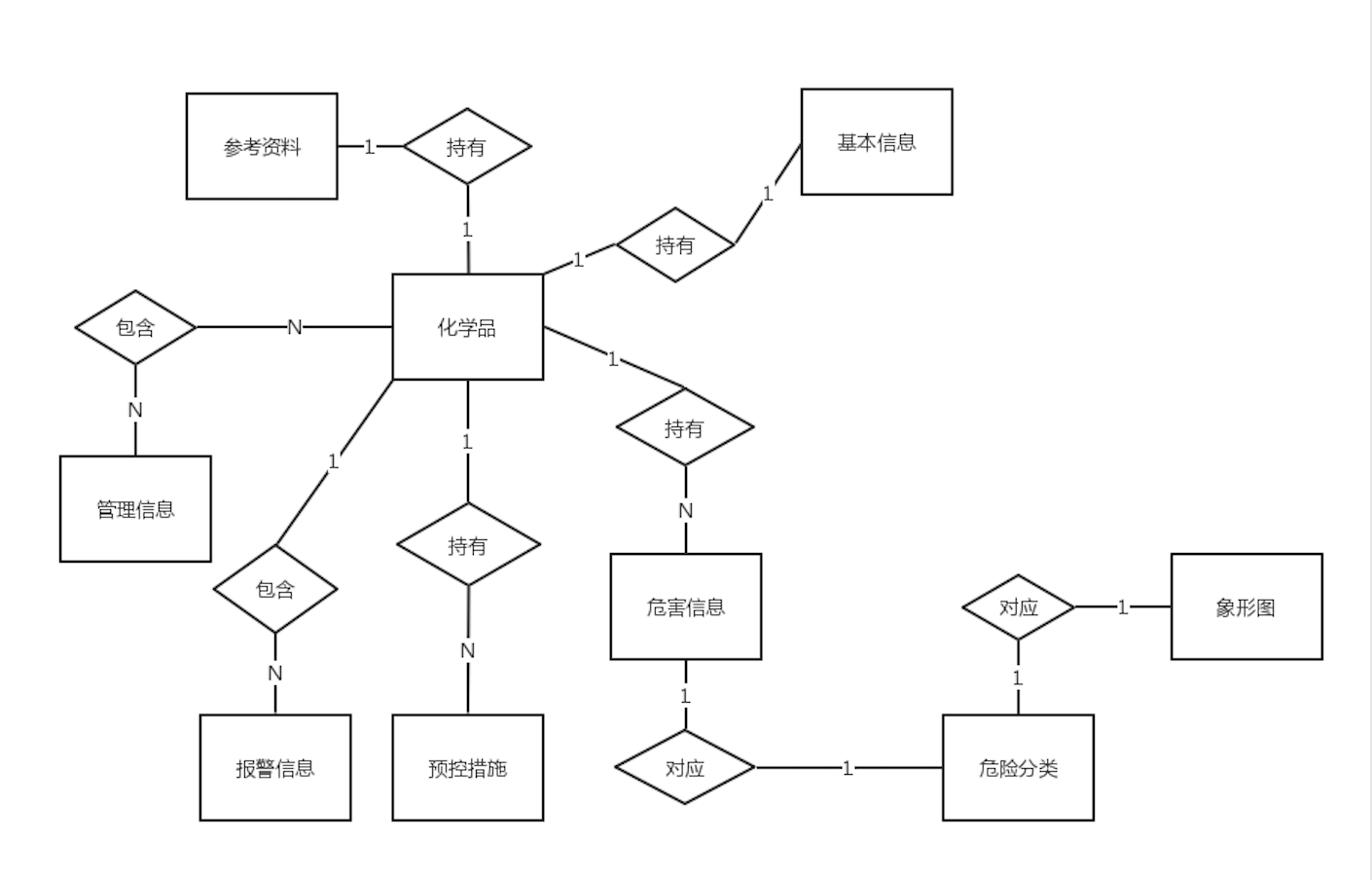


图4.3 系统E-R图

### 4.4.2 数据库表

根据E-R图，设计系统关键数据库表及其关键属性如表4.2、4.3所示。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **代码** | **数据类型** | **注释** |
| 唯一性标识ID | id | int(11) | 主键、非空、自增 |
| 化学物质 | name | varchar(255) |  |
| 中文别名 | cn\_alia | varchar(255) |  |
| 英文名称 | en\_name | varchar(255) |  |
| 英文别名 | en\_alia | varchar(255) |  |
| CAS号 | cas | varchar(255) |  |
| 分子式 | formula | varchar(255) |  |
| 分子量 | weight | DOUBLE |  |
| 理化特性 | properties | TEXT | 存在状态、熔点、沸点、挥发性等 |
| 稳定性和反应性 | stability | TEXT |  |
| GHS危害性分类 | danger\_class | TEXT |  |
| 危险有害性概述 | overview | TEXT |  |
| 操作处置与储存 | word | TEXT |  |
| 接触控制/个体防护 | touch\_control | TEXT |  |
| 急救措施 | first\_aid | TEXT |  |
| 泄露应急处理 | leakage | TEXT |  |
| 废弃处置 | disposal | TEXT |  |

表4.2 化学品基本信息及危害信息表关键属性

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **代码** | **数据类型** | **注释** |
| 唯一性标识ID | id | int(11) | 主键、非空、自增 |
| 化学品ID | chemical\_id | int(11) | 外键 |
| 备注信息 | remarks | TEXT |  |
| 位置信息 | position | varchar(255) |  |

表4.3 报警信息表关键属性

## 4.5 系统包图

项目采用前后端分离。整体的体系结构采用分层模式，完成了功能上的内聚，这提升了各层的易理解性和可维护性。化学品危险信息流转识别系统的模块包图如图4.4所示，整体上描述了各个包之间的详细关系。

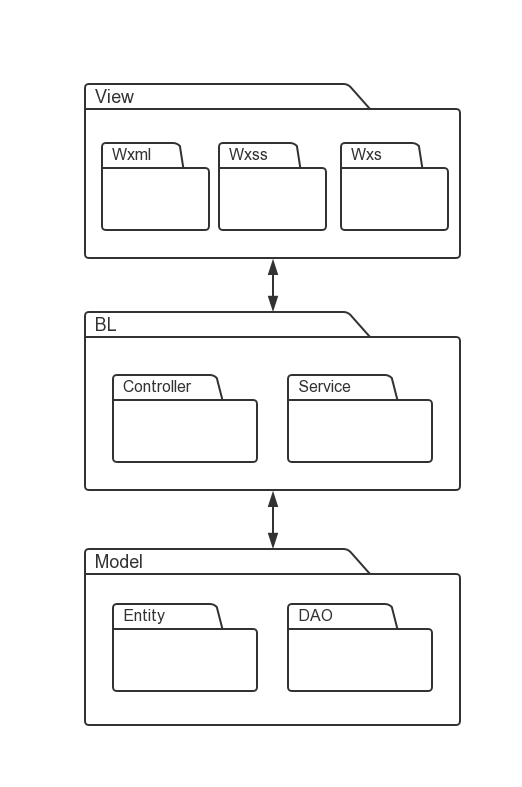


图4.4 系统包图

View层负责与界面请求的交互功能实现。Controller层主要是Controller类，负责处理请求，控制业务模块流程，Controller会调用Service层中的Service类来处理业务模块的应用逻辑。持久层包括负责数据处理的DAO以及对数据封装的Entity包。Service处理业务后会调用DAO类，DAO在持久层处理请求并返回结果，直接对Entity中的对象操作即可达到同样的数据库增删查改操作。

## 4.6 系统详细类图

化学品危险信息流转识别系统中涉及了众多的类，各个类之间协作紧密。这里以查询化学品的详细类图为例，分析业务逻辑、内部接口调用，由此阐述系统的详细类图设计。详细类图如图4.5所示，查询化学品功能模块中每个类职责清晰，只关注自己的业务逻辑。当用户发起查询请求时，ChemicalController处理查询操作，具体调用ChemicalService，返回查询的化学品信息。

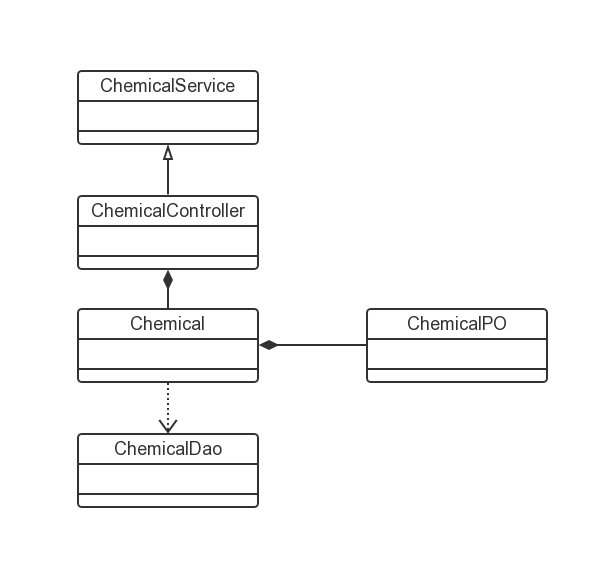


图4.5 详细类图

## 4.7 系统时序图

针对系统中的查询化学品、查询历史详情、报警和查看报警信息功能，分析系统的处理流程和用户交互的过程。

### 4.7.1 扫描二维码查询流程

当用户使用扫描二维码查询化学品时，系统显示取景框界面，用户将二维码放至取景框内，系统会自动扫描二维码，调用查询化学品服务得到对应的化学品信息，然后将查询结果返回给用户，如图4.6所示。

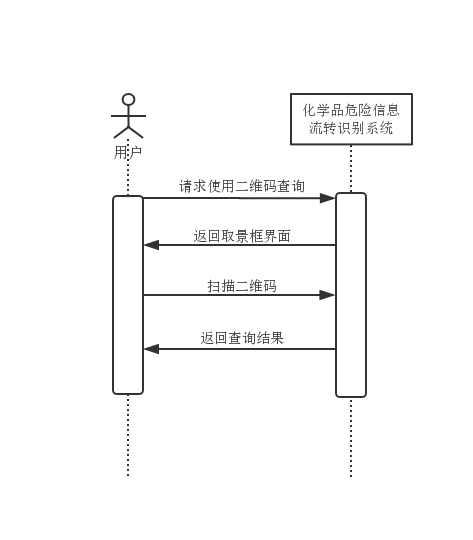


图4.6 扫描二维码查询时序图

### 4.7.2 根据名称查询流程

用户输入化学品名称并选择使用名称查询后，系统调用查询化学品服务得到对应的化学品信息，然后将查询结果返回给用户，如图4.7所示。

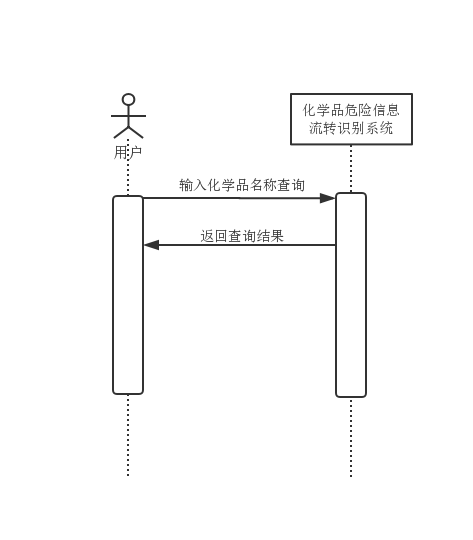


图4.7 根据名称查询时序图

### 4.7.3 根据CAS码查询流程

用户输入CAS码并选择使用CAS码查询后，系统调用查询化学品服务得到对应的化学品信息，然后将查询结果返回给用户，如图4.8所示。

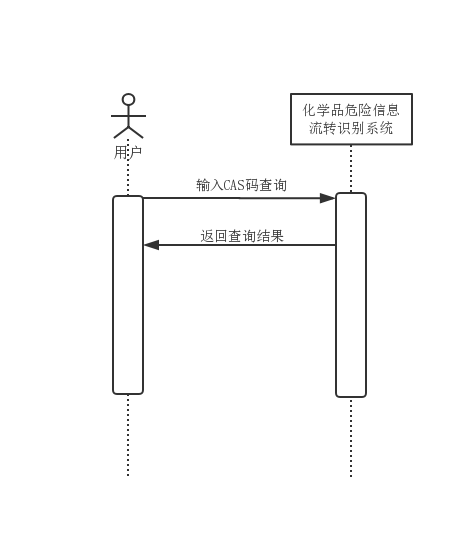


图4.8 根据CAS码查询时序图

### 4.7.4 查询历史详情流程

用户查询化学品信息后，系统会调用历史信息服务来生成历史信息记录，然后，历史信息服务会在数据库中插入一条新的记录。用户选择查看历史记录时，会调用历史信息服务，系统显示所有历史信息记录。当用户点击查看历史记录详情时，系统显示该历史记录对应的化学品信息，如图4.9所示。

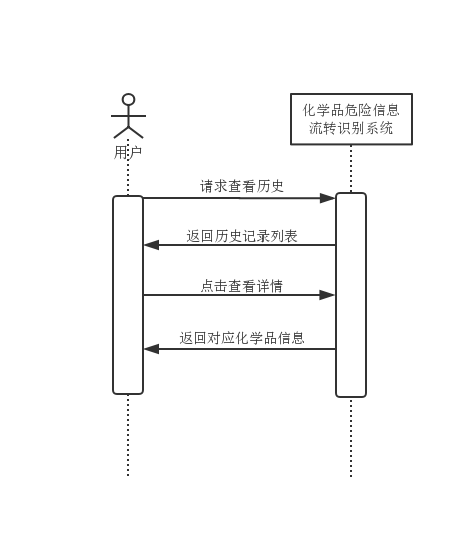


图4.9 查询历史详情时序图

### 4.7.5 报警流程

用户上传报警信息后，系统会调用报警服务来生成报警信息，然后，报警服务会在数据库中插入一条新的记录，系统返回上传成功消息，如图4.10所示。

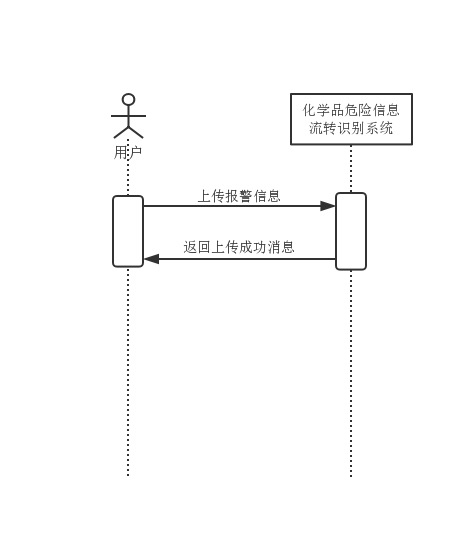


图4.10 报警时序图

### 4.7.6 查看报警信息流程

当管理员查看报警信息时，系统调用报警服务显示所有报警信息，如图4.11所示。

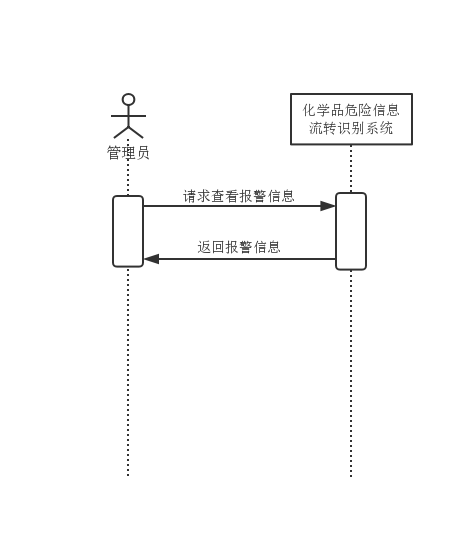


图4.11 查看报警信息时序图