# 系统需求分析

## 1.1业务流程分析

在进行详细的需求分析之后，确定了本系统应该有管道巡线人员、系统管理员、抢险人员等用户，在此基础上我们得出系统应该具备的功能何业务流程，主要包括以下几点：

（1）燃气失效系统网站主要在PC端浏览器访问

（2）系统根据不同权限应该具有失效事件的不同阶段处理能力，如：事件上报，事件指派，事件完成提交等

（3）系统的各项数据可以通过Excel格式导入或导出

（4）针对数据进行统计分析，图表分析

## 1.2功能性需求分析

燃气管道失效抢险系统是基于B/S体系结构模型的系统，采用前后端分离的模式进行开发。本系统主要分为地图模块、事件管理模块、人员管理模块、任务中心模块、信息管理模块、登录注册模块。各模块的功能阐述具体如下：

（1）地图模块：主体功能是对地图上管线的展示，用户可通过点击地图上的管线点进行事件的编辑上报，或者点击报警的事件进行事件的审核处理。其他功能包括巡线工的位置显示，失效事件的提示列表等

（2）事件管理模块：事件管理模块主要分为事件查询和事件处理两部分，事件查询可根据事件的状态、位置等条件进行事件检索；事件处理可以让管理人员对事件的状态进行修改，对事件后续抢险工作进行指派，并且可以对该事件的部分信息作补充和修改

（3）人员管理模块：该模块主要用于管理管道巡线人员和抢险人员。包括以下功能：新建管道巡线人员和抢险人员，编辑和删除管道巡线人员和抢险人员的信息，根据人员类型进行过滤和搜索

（4）任务中心模块：该模块主要用于管理抢险任务。包括以下功能：根据现有的失效事件新建任务并指派给相应的抢险人员，查看任务列表并对任务进行编辑和删除操作，根据任务状态进行过滤和搜索

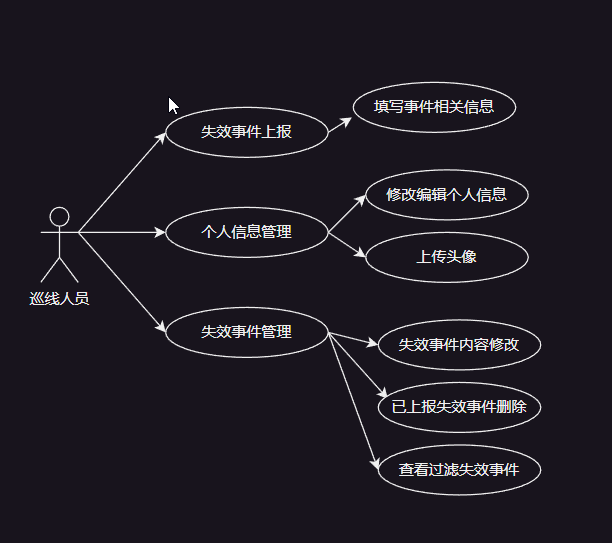
（5）信息管理模块：该模块主要用于对各用户的个人信息进行填写，包含用户头像、昵称、电话号码、邮箱等信息。

（6）登录注册模块：该模块主要实现用户登录、注册功能，登录注册过程中可以选择不同的身份进行操作。

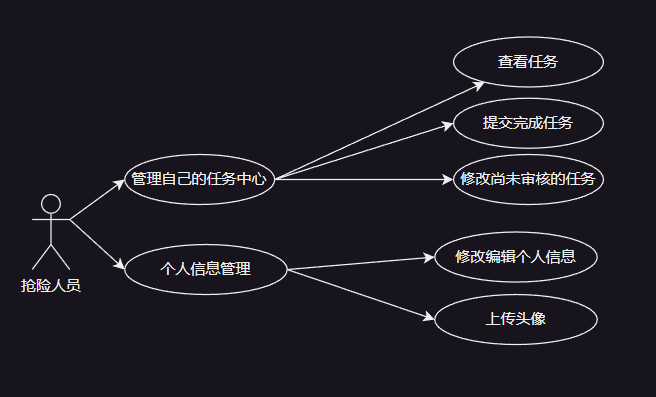
## 1.3用例分析

根据对燃气失效抢险系统的实际应用功能进行需求分析

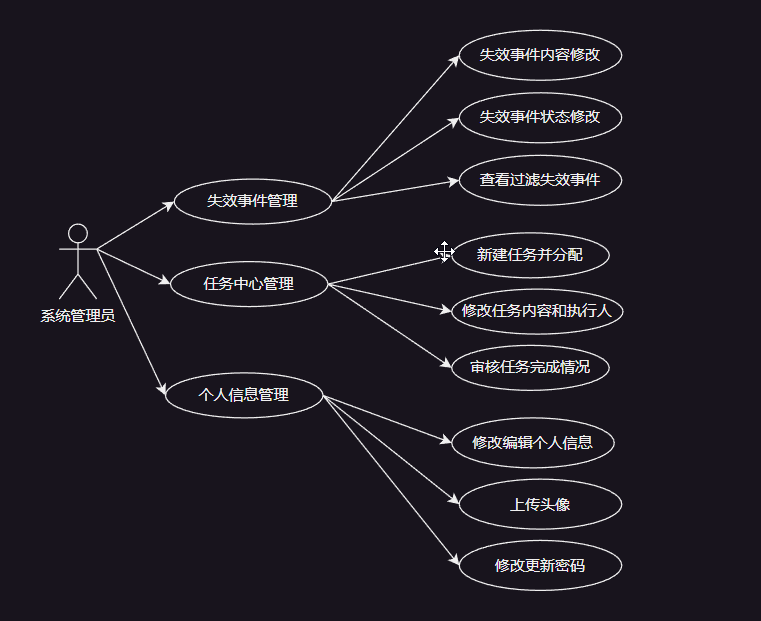
（1）本系统的巡线人员用户用例图如下所示。



（2）关于抢险人员的用例模型如下图所示。



（3）关于系统管理员的用例模型如下图所示。



## 1.4非功能性需求分析

（1）用户交互需求：

考虑到使用该系统的用户不一定对计算机操作熟练，所以在功能的设计上需要尽量简单明了，增加人性化的帮助工具，以及符合大众审美的界面，将交互页面设计成人人可用的实用型页面

（2）软件环境需求：

本系统的软件环境需求主要包含操作系统平台，数据库平台，软件开发平台，服务器平台，地图显示平台等。

操作系统平台采用电脑上现有的windows10操作系统，该系统的在大众面上受众最广，属于最常见的操作系统。

数据库平台选用SQLyog作为数据库的图形化界面，使用mysql语言来完成数据库的搭建

软件开发平台选用比较轻量好用的vscode软件，vscode得益于它庞大的插件商店，可以自由搭配出最适合自己开发习惯的工具。我们主要使用的开发语言为JavaScript脚本语言，前端使用vue框架，后端使用nodeJs的koa框架

服务器平台使用高性能的web服务器：nginx，作为一款轻量级的服务器，它的配置简单，资源消耗少，并发能力好，稳定且功能丰富，在日常开发中也是非常常见的一款服务器。

地图显示平台我们采用Arcgis api搭配天地图来显示我们的管线地图。

（3）性能需求：

本系统的性能要求是，符合正常软件性能标准，在保证稳定的同时还要考虑兼容性设计和对异常情况的处理，对程序中的关键逻辑部分增加代码注释，以备后续维护和功能扩展

（4）硬件环境要求：

本系统采用经典的B/S架构，将系统部署在服务器上，用户通过浏览器端随时随地访问。浏览器兼容IE和chrome浏览器，服务器端的内存理应不低于2048M

# 系统设计

## 1.1系统架构设计

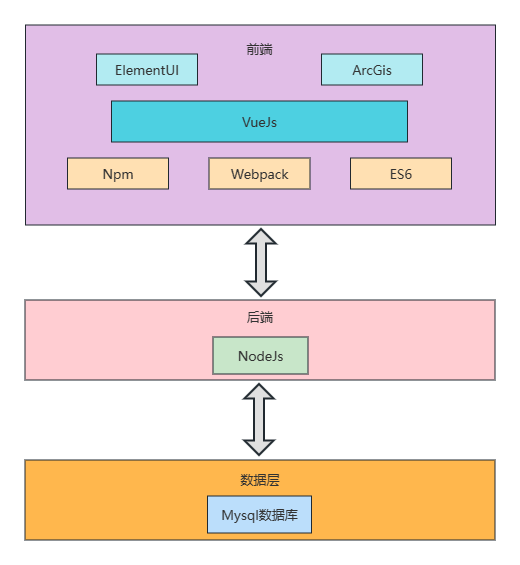
设计燃气管道失效抢险系统时采用B/S架构，这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用；同一般B/S架构系统一样，系统分为前端和后端，整个系统依赖Node环境。

前端采用vue和arcgis实现界面框架以及地图功能，使用ElementUI组件库来提升开发效率，使用Npm对依赖包进行管理，使用Webpack打包工具来进行前端部分的打包与性能优化

后端采用node.js的Koa框架和mysql实现数据库操作与数据处理。

Nginx服务器是一款轻量级的web服务器，且拥有占用资源少，稳定，并发能力强的特点，资源配置文件也十分简单。因此服务器端采用Nginx服务器来对项目代码进行部署运行。

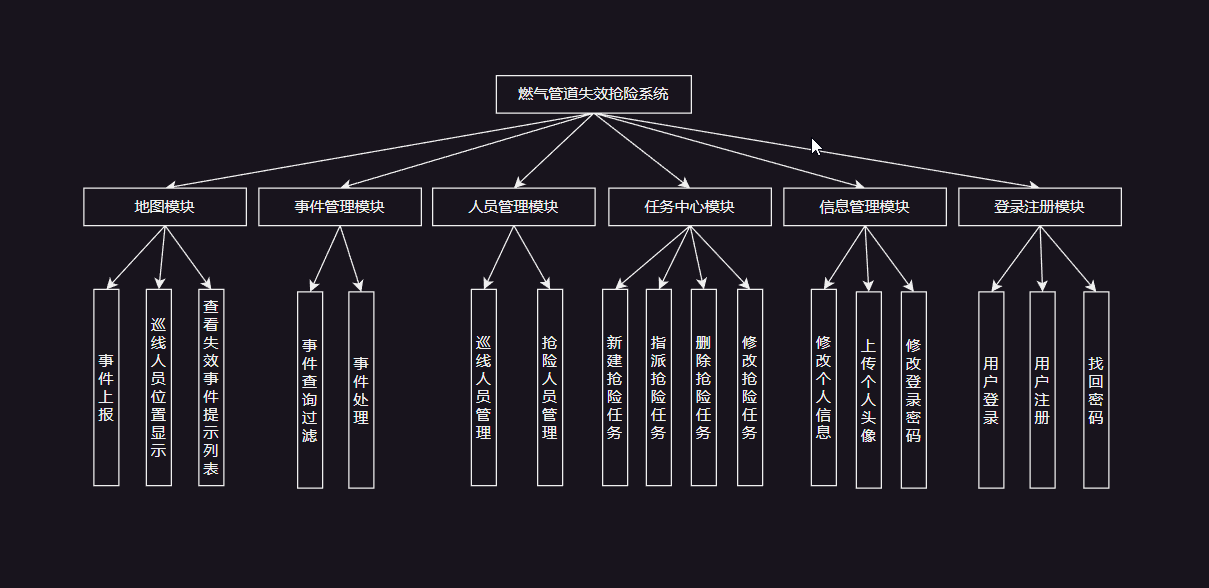
系统总体架构如图：



**图 1系统总体架构图**

## 系统功能结构

燃气管道失效抢险系统主要负责对燃气失效事件的上报、审核、指派、处理等流程，系统结构图如下：



**图 2系统功能结构图**

## 系统设计原则

为了更好的确保燃气管道失效抢险系统的设计高效性，在本文中提前为系统约定了设计原则来对开发进行约束规范，在开发过程中严格遵守相关规范，提高设计效率和业务处理时间，参考相关案例设计，我们给出四个方面，分别是：实用性原则、安全性原则、易维护性原则、易操作性等，下面是针对这几个方面的详细阐述：

1. 实用性原则：设计符合实际应用需求的功能和流程；采用成熟、可靠的技术进行开发，保证系统具备稳定性和可用性；设计合理的权限管理和数据保护机制，确保系统数据安全和用户隐私保护；系统界面友好、操作简单，提高用户满意度和使用效率；站在用户的角度理解需求，增加用户的使用体验是重中之重。
2. 安全性原则：本系统的安全性原则主要考虑管线数据的问题，由于采用真实的燃气公司管道数据，在部署地图服务时，需要格外考虑安全问题，如：规范接口调用的参数、数据库表的合理设计、将ArcGis Server地图服务部署在与后端不同的服务器上来规避部分风险。
3. 易维护性原则：采用模块化设计，将不同功能模块的代码分开，减少耦合性，便于单独完善和维护模块代码；采用标准化的接口和协议，方便到了开发后期不可避免的替换和扩展；代码注释详细、规范，便于其他开发人员以及自己在后期对代码进行维护；提供完善的错误处理和日志记录机制，便于开发人员快速定位和解决问题；定期进行系统维护和升级，确保系统稳定性和安全性。
4. 易操作性原则：界面设计简单明了，便于用户操作；提供清晰的操作指南和帮助文档，以便用户能够快速上手；尽可能使用图形化界面、按钮、下拉菜单等方式，减少用户对系统命令的记忆负担；合理设置默认值、提示信息等，提高用户使用系统的效率和舒适度。

## 系统各模块的详细设计

依据燃气管道失效抢险系统的需求分析，我们把系统划分成四个主要部分，分别是地图模块、事件管理模块、人员管理模块、任务中心模块。下面我们分别从这四个部分对燃气管道失效抢险系统的设计进行说明。

**1.2.1 地图模块**

地图模块是整个系统的核心部分，它需要支持以下功能：

（1）管线的展示：根据管线的坐标信息，将管线在地图上展示出来，用户可以通过地图放大、缩小等操作查看管线的具体信息。

（2）事件上报：用户可以通过点击地图上的管线点进行事件的编辑上报，包括事件的类型、位置、描述等信息。

（3）事件审核：管理员可以点击地图上的报警事件进行审核处理，包括查看事件详细信息、修改事件状态、指派抢险任务等操作。

（4）巡线工位置显示：系统需要实时获取巡线工的位置信息，并在地图上展示其位置。

（5）失效事件提示列表：系统需要在地图上展示当前存在的失效事件列表，并根据事件的紧急程度进行排序和提示。

为了实现这些功能，我们使用ArcGis API来展示管线和位置信息，并通过前端Vue.js框架实现事件上报和审核功能。

**1.2.2 事件管理模块**

对于事件管理模块，我们需要设计以下几个功能：

（1）事件查询：可以根据事件的状态、位置、时间等条件进行事件检索。用户可以根据自己的需求对事件进行筛选，筛选结果将以列表的形式呈现出来。同时，还需要支持对筛选结果进行导出操作，方便用户进行数据统计分析以及保存。

（2）事件处理：主要用于对事件的状态进行修改，对事件后续抢险工作进行指派，并且可以对该事件的部分信息作补充和修改。需要考虑到用户的操作体验，设计简单易懂的操作流程，并且提供必要的提示信息，以便用户更好地完成事件处理操作。

（3）事件审核：用于审核已经上报的事件。管理员可以在地图模块或事件查询子模块中查看待审核的事件，并对事件进行审核。审核结果将会对事件状态进行修改，同时还需要支持对审核记录的查询和导出，方便用户进行数据统计分析以及保存。

**1.2.3 人员管理模块**

人员管理模块是系统的重要组成部分，提供对巡线和抢险人员的管理。该模块应包括以下功能：

（1）新建巡线和抢险人员：管理员可以通过该功能新增巡线和抢险人员。新增人员需要填写姓名、联系方式、所在地区等基本信息，以及人员类型、身份证号码等详细信息。

（2）编辑和删除人员信息：管理员可以对巡线和抢险人员的信息进行修改和删除。修改信息包括基本信息和详细信息，例如人员类型、身份证号码、专业技能等。删除人员信息时需要进行二次确认，以避免误操作。

（3）过滤和搜索：管理员可以根据人员类型进行过滤和搜索，以便快速查找特定类型的人员。

（4）导入和导出：管理员可以通过导入和导出功能批量导入和导出巡线和抢险人员信息。导入时需要指定导入文件的格式和路径，导出时可以选择要导出的信息内容和格式。

**1.2.4 任务中心模块**

任务中心模块是燃气管道失效抢险系统中非常重要的一部分，它主要用于管理抢险任务，包括以下功能：

（1）新建任务并指派给相应的抢险人员：管理员可以根据已有的失效事件和管道巡检信息，新建相应的抢险任务，并指派给相应的抢险人员。在新建任务时，管理员需要填写任务的基本信息，例如任务名称、任务描述、任务地点、任务等级、预估时间等，同时需要指定负责该任务的抢险人员。

（2）查看任务列表并对任务进行编辑和删除操作：管理员可以查看系统中所有的任务，并可以对任务进行编辑和删除操作。对于已经完成的任务，管理员可以将任务状态改为已完成，并填写相关的任务完成信息。

（3）根据任务状态进行过滤和搜索：管理员可以根据任务的状态，例如待处理、处理中、已完成等状态，进行过滤和搜索操作，以便快速找到需要处理的任务或查看已经完成的任务。

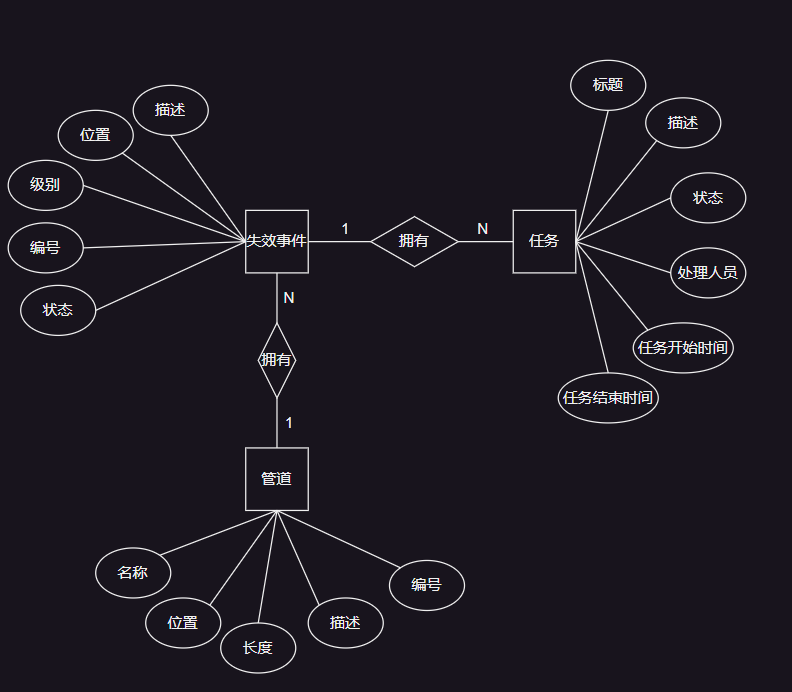
（4）统计任务情况：管理员可以通过任务中心模块，对系统中的任务情况进行统计，例如查看某段时间内任务的完成情况、任务的平均处理时间等统计信息。

**1.3 数据库设计**

**1.3.1 数据关系设计**

基于我们的需求分析和系统功能设计，我们可以清晰直观的认识到系统实体之间的关系，我绘制了全体E-R实体关系图如下图所示：

**图 3 全体E-R实体关系图**



**1.3.2 数据库表设计**

**表 1.1 用户表(user)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 是否允许为空 | 描述 |
| id | int | 是 | 否 | 用户id，自增长 |
| username | varchar(20) | 否 | 否 | 用户名 |
| password | varchar(20) | 否 | 否 | 密码 |
| actor\_url | varchar(255) | 否 | 是 | 头像存放地址 |
| email | varchar(50) | 否 | 是 | 邮箱 |
| phone | varchar(20) | 否 | 是 | 电话号码 |
| role | int | 否 | 否 | 用户角色，1为管道巡线人员，2为抢险人员，3为管理员 |

**表 1.2 事件表(events)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 是否允许为空 | 描述 |
| id | int | 是 | 否 | 事件id，自增长 |
| title | varchar(50) | 否 | 否 | 事件标题 |
| description | varchar(255) | 否 | 是 | 事件描述 |
| location | varchar(50) | 否 | 是 | 事件位置 |
| longitude | decimal(10,6) | 否 | 是 | 事件位置经度 |
| latitude | decimal(10,6) | 否 | 是 | 事件位置纬度 |
| status | int | 否 | 否 | 事件状态，0为未处理，1为已处理，2为已完成 |
| submit\_time | datetime | 否 | 否 | 事件提交时间 |
| handler\_id | int | 否 | 是 | 事件处理人员id，若未处理则为null |
| handler\_time | datetime | 否 | 是 | 事件处理时间，若未处理则为null |
| manager\_id | int | 否 | 是 | 事件管理人员id，若未指派则为null |
| pipeline\_id | int | 否 | 否 | 事件所属管道id |
| level | int | 否 | 否 | 事件级别，1为一般，2为重要，3为紧急 |

**表 1.3 管道表(pipelines)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 是否允许为空 | 描述 |
| id | int | 是 | 否 | 管道id，自增长 |
| name | varchar(50) | 否 | 否 | 管道名称 |
| description | varchar(255) | 否 | 是 | 管道描述 |
| length | decimal(10,2) | 否 | 否 | 管道长度 |
| start\_location | varchar(50) | 否 | 否 | 管道起点位置 |
| end\_location | varchar(50) | 否 | 否 | 管道终点位置 |

**表 1.4 任务表(task)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 是否允许为空 | 描述 |
| id | int | 是 | 否 | 任务ID |
| event\_id | int | 否 | 否 | 事件ID |
| title | varchar(255) | 否 | 否 | 任务标题 |
| description | varchar(255) | 否 | 是 | 任务描述 |
| handler\_id | int | 否 | 否 | 处理人员ID |
| handler\_name | varchar(50) | 否 | 否 | 处理人员姓名 |
| handler\_type | int | 否 | 否 | 处理人员类型，1表示巡线人员，2表示抢险人员 |
| status | int | 否 | 否 | 任务状态，1表示待处理，2表示处理中，3表示已完成 |
| start\_time | datetime | 否 | 否 | 任务开始时间 |
| end\_time | datetime | 否 | 是 | 任务结束时间 |
| created\_time | datetime | 否 | 否 | 创建时间 |
| updated\_time | datetime | 否 | 否 | 更新时间 |