目录

[1. 绪论 2](#_Toc3391)

[1.1 研究背景 2](#_Toc21041)

[1.2 国内外研究现状 2](#_Toc28586)

[1.3 选题的意义 3](#_Toc26752)

[2 相关技术介绍 3](#_Toc1963)

[2.1 B/S架构介绍 3](#_Toc21712)

[2.2 前端技术介绍 4](#_Toc22114)

[2.3 ArcGIS地图api介绍 4](#_Toc7132)

[2.4 后端技术介绍 5](#_Toc7500)

[2.5 数据库介绍 5](#_Toc21014)

[2.6 web服务器介绍 6](#_Toc15444)

[3 系统需求分析 6](#_Toc2042)

[3.1业务流程分析 6](#_Toc22878)

[3.2功能性需求分析 6](#_Toc3552)

[3.3用例分析 7](#_Toc25915)

[3.4非功能性需求分析 9](#_Toc29619)

[4 系统设计 10](#_Toc28967)

[4.1系统架构设计 10](#_Toc2792)

[4.1.1系统功能结构 10](#_Toc8506)

[4.1.2系统设计原则 11](#_Toc9647)

[4.2系统各模块的详细设计 12](#_Toc11134)

[4.3 数据库设计 14](#_Toc25657)

[5 系统实现 16](#_Toc4222)

[5.1系统登录页面 16](#_Toc10620)

[5.2系统首页地图界面 17](#_Toc24974)

[5.3事件统计页面 17](#_Toc23271)

[5.4人员管理页面 18](#_Toc18976)

[5.5任务中心页面 18](#_Toc27358)

[6系统测试 19](#_Toc31960)

[6.1测试目标 19](#_Toc27644)

[6.2 测试计划和要点 19](#_Toc27208)

[6.3 功能测试及内容 20](#_Toc7014)

[6.4测试用例及结果 21](#_Toc5751)

[7 总结与展望 22](#_Toc693)

[致谢 23](#_Toc19352)

# 绪论

## 研究背景

燃气作为高效清洁能源被广泛使用，从20世纪80年代开始，我国城市燃气事业开始迅速发展，并在大中型城市开始形成燃气输配管网，根据住房和城乡建设部 2019 年城乡建设统计年鉴显示，截至2019年底，全国天然气管网长度达78.3万公里。

由于城镇燃气管道运行环境复杂，受管道老化、第三方施工作业、管道环境因素等影响，近年来各燃气公司管网泄漏事故时有发生，常造成人员伤亡、财产损失和不良社会影响。建立具有燃气管道泄漏监测预警、智能辅助决策、应急处置分析等功能的智能分析与应急处置系统，对提升燃气企业信息化、智能化管理水平，支撑燃气管网安全运营管理、保障城市居民生命和财产安全等具有重要意义。

## 1.2 国内外研究现状

早在2002年，《上海煤气》就发表了一篇期刊《TUBIS^(TM)系统》[[[1]](#endnote-0)]来介绍美国 Optima 公司建立的 TUBIS^(TM)系统。它可成功地用于燃气输配管道网络等基础设施维护管理和风险管理。

而国内在2012年，也出现了有关燃气管道泄露的检测系统《基于无线传感网络的远程燃气管道泄漏检测系统》[[[2]](#endnote-1)]，其针对管道泄漏监测的应用场景,提出了一种管道泄漏远程实时检测的方案,其特点是结合GPRS网络和分簇结构的WSN实现远距离的数据传输。

在2019年甚至出现了使用神经网络算法对燃气管道故障进行识别诊断[[[3]](#endnote-2)]，准确率稳定在92.54%。

## 1.3 选题的意义

随着互联网的高速发展，各行业都在加快数字化进程，在有关石油燃气资源方面，我们也应该有着相应的创新。国内外燃气事故屡见不鲜，事故发生后，造成的人员以及财产损失巨大，同时对相关燃气管理公司的生存和发展带来危害。由于当下城镇燃气管道往往伴随着复杂多样的环境因素，为了避免燃气管道泄漏事故后产生的重大威胁，我们应该设计一套集泄露预警、应急处理、智能分析的系统，为应急抢修处置全流程提供可靠的决策依据。

# 2 相关技术介绍

## 2.1 B/S架构介绍

B/S架构是一种常见的软件系统架构，它的全称为Browser/Server架构，也称为Web架构。在这种架构中，客户端使用浏览器作为用户界面，而服务器提供应用程序的业务逻辑和数据访问功能。它的优点包括：

（1）跨平台：由于浏览器是跨平台的，所以不需要为不同操作系统开发不同的客户端软件，这大大简化了开发和维护的工作量。

（2）瘦客户端：浏览器作为客户端只需要负责用户界面，不需要执行复杂的业务逻辑，这样可以使客户端更加轻量级，减少了客户端的资源占用。

（3）易于维护：由于应用程序的业务逻辑和数据访问功能都在服务器端，因此可以通过集中式的方式进行管理和维护，这样可以提高系统的可维护性和可扩展性。

（4）安全性高：由于应用程序的业务逻辑和数据都在服务器端，因此客户端无法直接访问敏感数据，从而提高了系统的安全性。

（5）易于部署：B/S架构中的客户端只需要一个浏览器，因此系统的部署非常容易，只需要在服务器端安装应用程序和相关的Web服务器软件即可。

（6）易于更新：由于应用程序的业务逻辑和数据访问功能都在服务器端，因此可以通过升级服务器端软件来更新整个系统，这样可以减少客户端的升级工作量。

## 2.2 前端技术介绍

#### 2.2.1 Vue

Vue 是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

Vue.js是三大主流前端框架之一，关注视图层，容易上手，并拥有自己配套的第三方类库。基于数据驱动思想开发的javaScript框架，使用Vue框架开发可以大大减少繁杂的dom操作。数据驱动使得每一个HTML模块都可以高度内聚，通过给不同的HTML模块指定不同的数据模型，就可以实现前端的组件化开发方式。组件化可以降低系统各个功能的耦合性，并且提高了功能内部的聚合性。另外Vue框架还开发了一套自己的模板语言，采用虚拟dom的方式渲染HTML，这使得我们可以依赖Vue来做模板渲染，实现前后端分离的开发方式。

#### **2.2.2 ElementUI**

Element-UI是饿了么前端团队推出的一款基于Vue.js 2.0 的桌面端UI框架，一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的组件库，提供了配套设计资源，帮助网站快速成型。

## 2.3 ArcGIS地图api介绍

ArcGIS产品线为用户提供一个可伸缩的，全面的[GIS](https://baike.baidu.com/item/GIS/31541?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/ArcGIS/_blank)平台。ArcObjects包含了许多的可编程组件，从细粒度的对象（例如单个的几何对象）到[粗粒度](https://baike.baidu.com/item/%E7%B2%97%E7%B2%92%E5%BA%A6/53960096?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/ArcGIS/_blank)的对象（例如与现有ArcMap文档交互的地图对象）涉及面极广，这些对象为开发者集成了全面的GIS功能。

## 2.4 后端技术介绍

#### 2.4.1 Node.js

Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境，使用了一个事件驱动、非阻塞式 I/O 的模型，使其轻量又高效，其包管理器 npm，是全球最大的开源库生态系统。实现让 JavaScript 运行在服务端，让 JavaScript 成为与PHP、Python、Perl、Ruby 等服务端语言平起平坐的脚本语言。发布于2009年5月，由Ryan Dahl开发，实质是对Chrome V8引擎进行了封装。

Node对一些特殊用例进行优化，提供替代的API，使得V8在非浏览器环境下运行得更好。V8引擎执行Javascript的速度非常快，性能非常好，用于方便地搭建响应速度快、易于扩展的网络应用。Node使用事件驱动，非阻塞I/O 模型而得以轻量和高效，非常适合在分布式设备上运行数据密集型的实时应用。

**2.4.2 Koa框架**

Koa是一个基于Node.js的Web框架。Koa采用了ES6的语法，使用了async/await来处理异步请求，并使用了基于Promise的中间件机制。Koa的核心设计理念是中间件（Middleware），每一个HTTP请求都会经过一系列的中间件处理，这些中间件可以处理请求和响应，还可以将控制权转移给下一个中间件，从而实现更加灵活的请求处理。

## 2.5 数据库介绍

针对本系统，我们选择使用mysql数据库，mysql作为一款开源免费的数据库，风靡企业与个人开发者之间。经过长时间的市场检验以及大规模多类型的开发者实践操作，它的可靠性得到了证明。其性能也因为优化的查询语句、高效的索引、内存缓存等功能显得更加优秀，同时mysql还支持高并发的需求。且它的安装和配置十分简单，网络上很多优秀的教程和开发文档，安全性也因为其支持各种加密算法和数据备份恢复功能而有所保障。因此，基于mysql数据库，我们可以进行高效、稳定的管理系统数据。

## **2.6web服务器介绍**

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。因此本系统采用Nginx服务器来对项目代码进行部署运行。

# 3 系统需求分析

## 3.1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和业务流程，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

## 3.2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

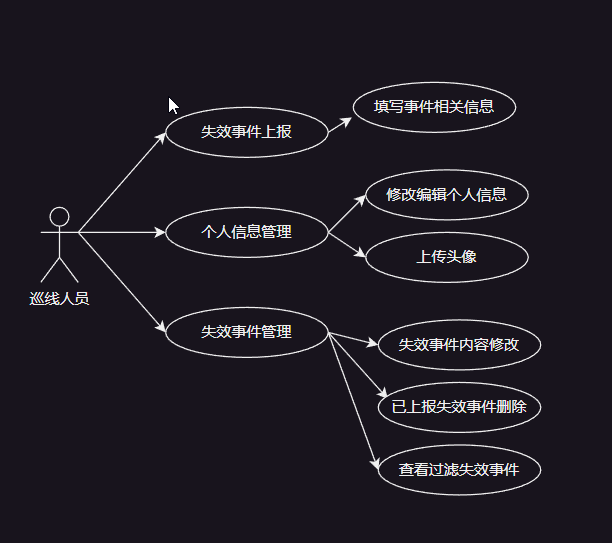
（5）信息管理模块：该模块主要用于对各用户的个人信息进行填写，包含用户头像、昵称、电话号码、邮箱等信息。

（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。

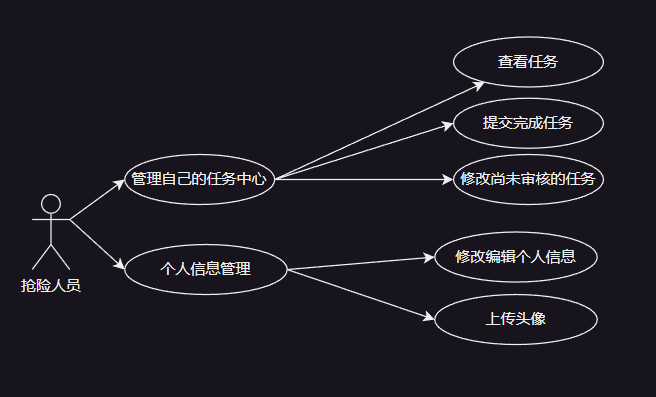
## 3.3用例分析

根据对燃气失效抢险系统的实际应用功能进行需求分析

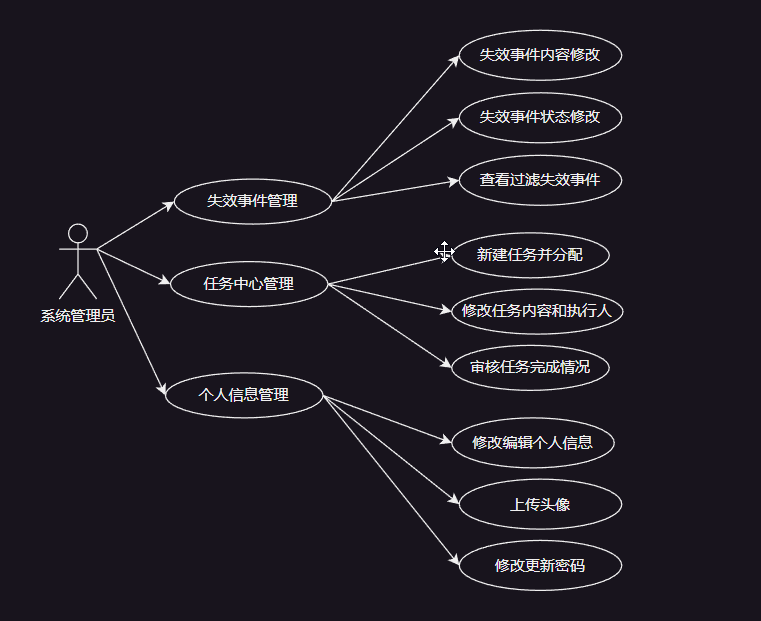
（1）本系统的巡线人员用户用例图如下所示。



（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。



## 3.4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

# 4 系统设计

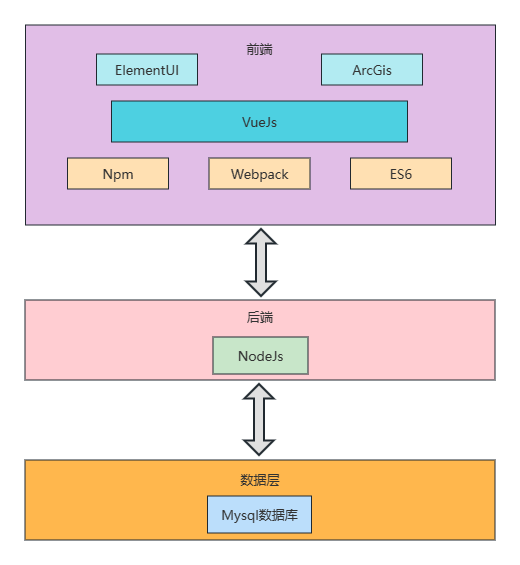
## 4.1系统架构设计

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。

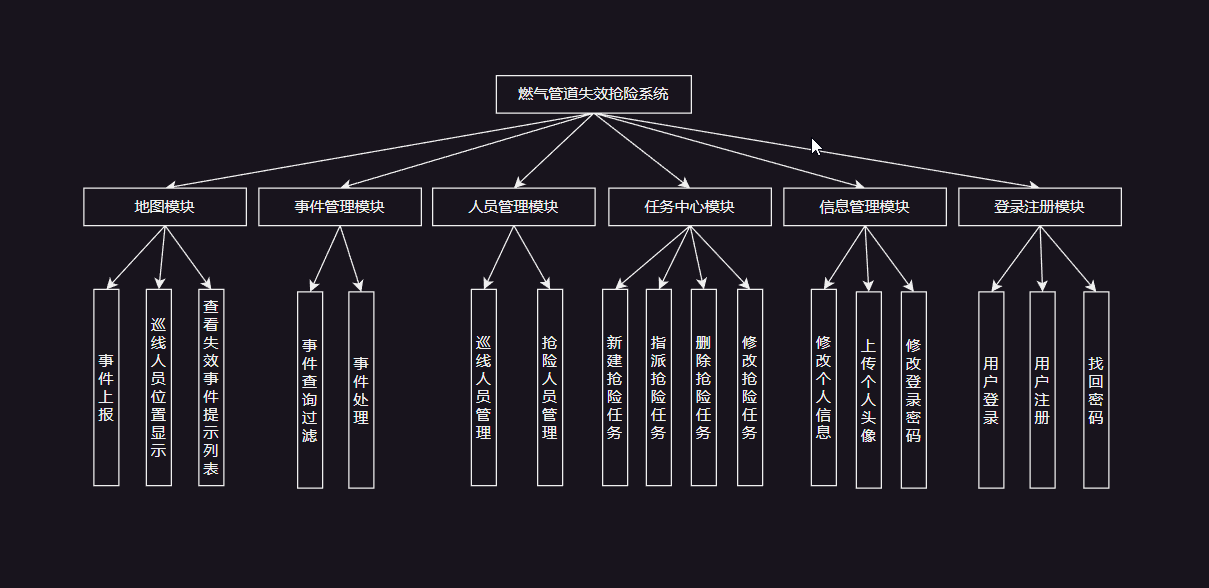
系统总体架构如图：



**图 1系统总体架构图**

## 4.1.1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**

## 4.1.2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。

## 4.2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。

**4.2.1 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。

**4.2.2 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。

**4.2.3 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。

**4.2.4 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。

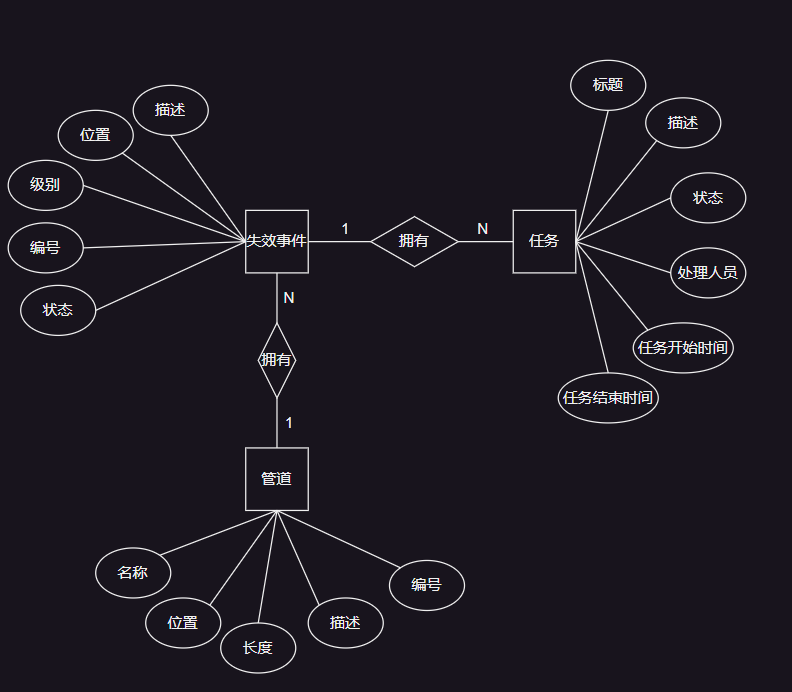
（4）统计任务情况：管理员可以通过任务中心模块，对系统中的任务情况进行统计，例如查看某段时间内任务的完成情况、任务的平均处理时间等统计信息。

**4.3 数据库设计**

**4.3.1 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**



**4.3.2 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 |

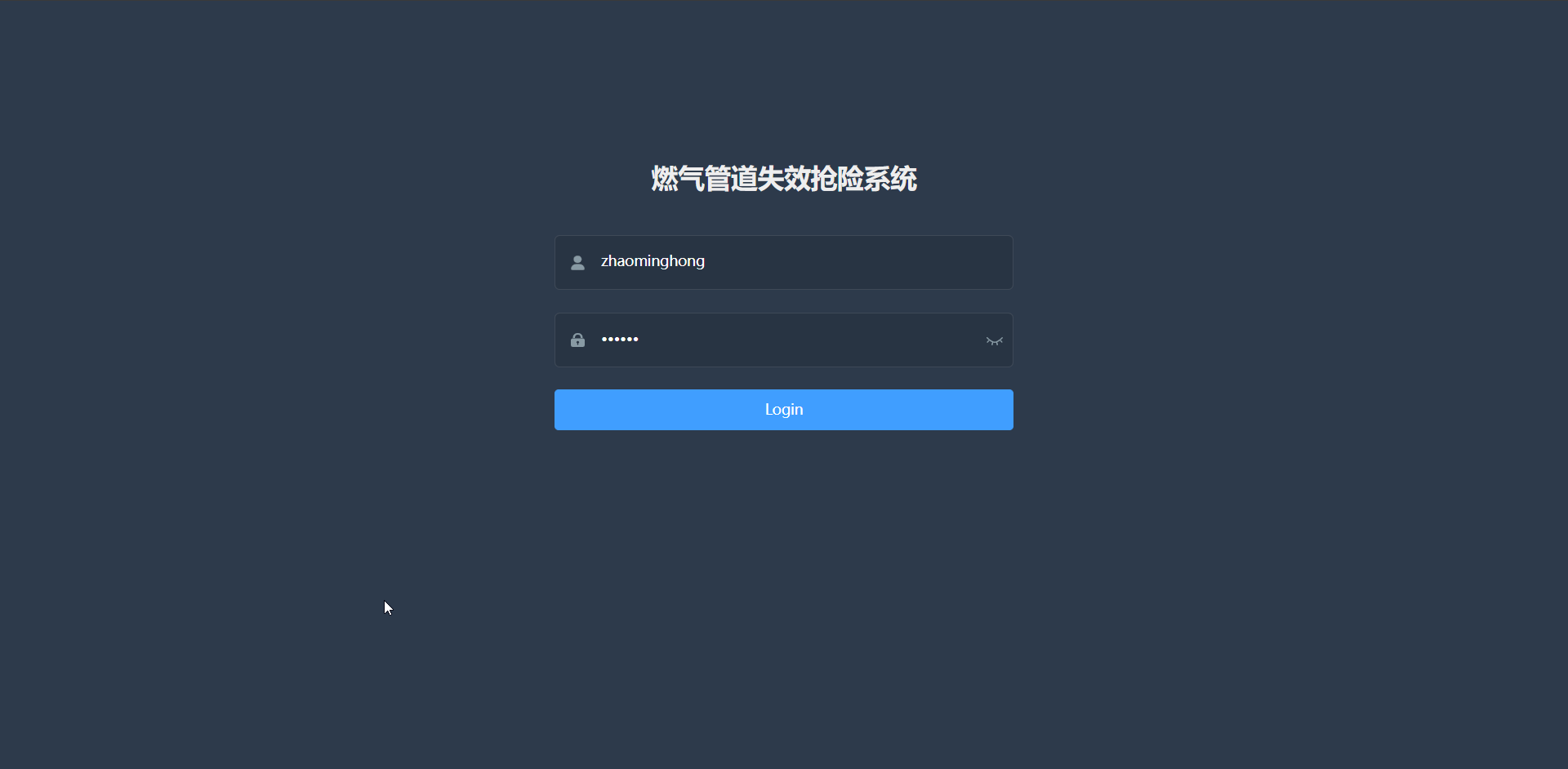
**表 1.5 人员表(personnel)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 是否允许为空 | 描述 |
| id | int | 是 | 否 | 主键 |
| name | varchar(50) | 否 | 否 | 姓名 |
| identity | int | 否 | 否 | 身份 |
| identityNo | varchar(18) | 否 | 否 | 身份证号码 |
| telePhoneNo | varchar(20) | 否 | 否 | 手机号码 |
| email | varchar(50) | 否 | 是 | 邮箱 |
| actor | varchar(255) | 否 | 是 | 角色 |

# 5 系统实现

## 5.1系统登录页面

**图 4登录页面**



## 5.2系统首页地图界面

**图 5首页界面**



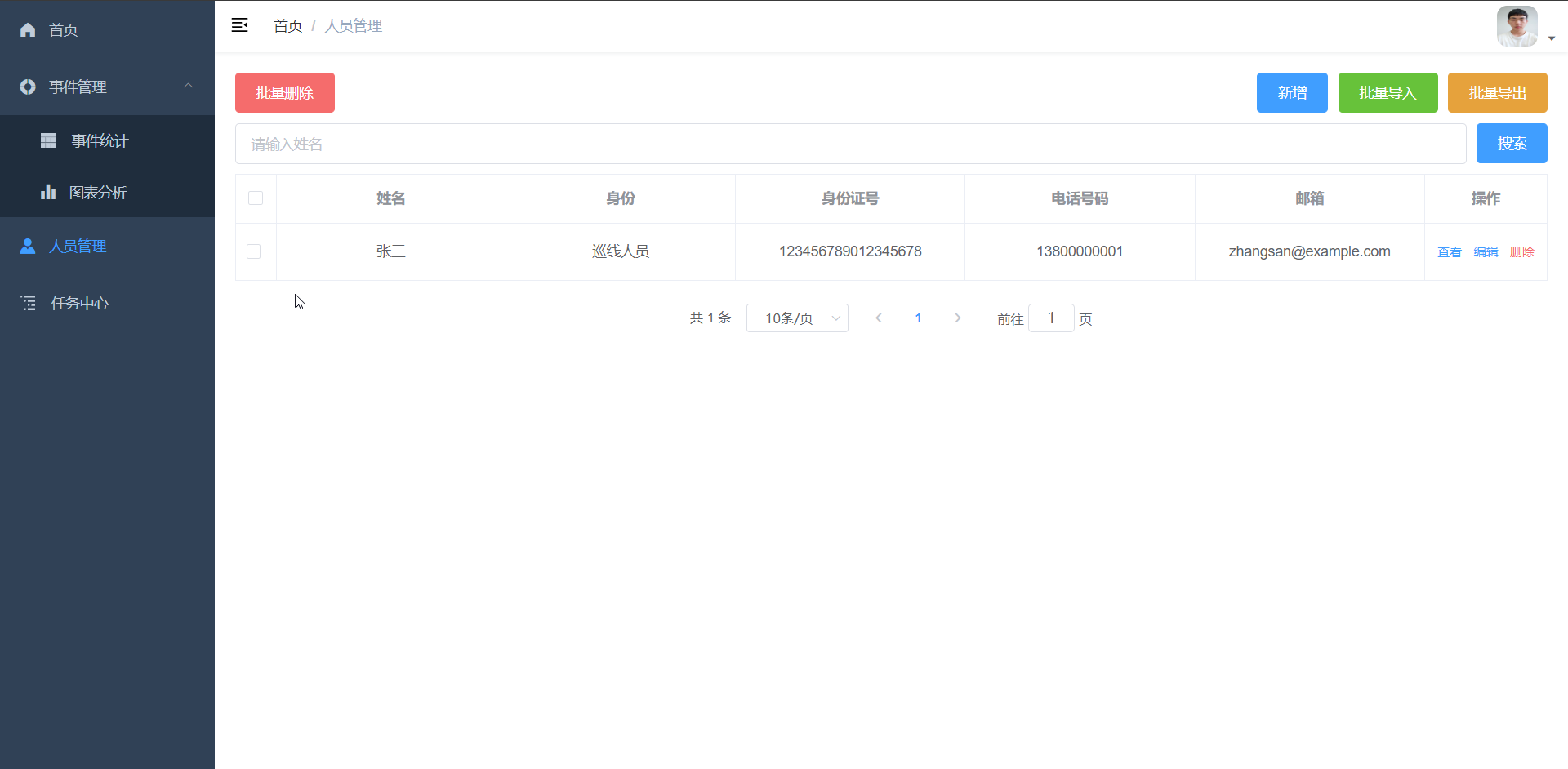
**5.3事件统计页面**

**图 6事件统计界面**



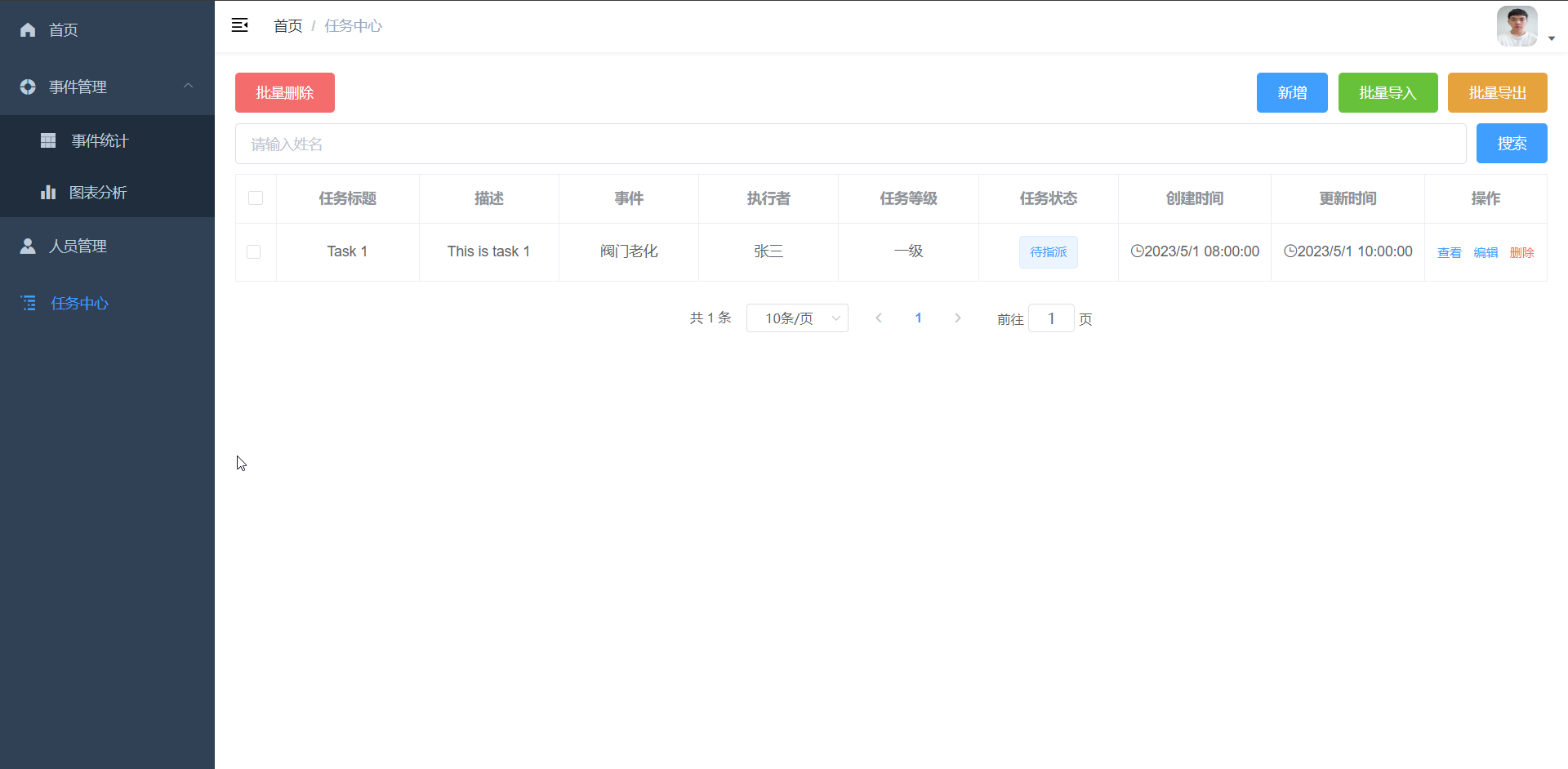
**5.4人员管理页面**

**图 7人员管理界面**



**5.5任务中心页面**

**图 8任务中心界面**



# 6系统测试

### 6.1测试目标

本系统在开发完成之后，为保证系统的正常工作流程能够完成，进行了主要功能的测试。该系统的功能结构较为简单，结构清晰明了，故采用黑盒测试的方式，通过页面上输入不同的数据来进行测试，数据主要是采用各项边界值来检验系统是否存在bug。通过不断修复bug以达到系统能够在服务器上稳定运行的目的。

### 6.2 测试计划和要点

（1）测试用户登录注册功能是否正常，能否通过新注册的用户名和密码登录成功，并且获取到用户如头像等信息

（2）测试浏览器显示的页面是否正常，会不会因为浏览器的不同出现样式丢失、样式扭曲、事件失效等情况

（3）测试后台数据，观察在后台手动修改数据，页面前端能否正常显示已修改的数据和页面

### 6.3 功能测试及内容

#### 6.3.1系统登录与身份验证测试

（1）测试用户能否通过登录页面登录系统并显示正确的用户信息

（2）测试输入错误的用户名或密码时，系统是否能够正确提示用户登录失败

#### **6.3.2事件上报和处理测试**

（1）测试管道巡线人员能否上报事件，并确保事件的相关信息能够正确显示在事件管理模块中

（2）测试系统管理员是否能够正确处理事件，包括指派任务和修改事件状态等操作

（3）测试抢险人员是否能够接收指派的任务，并将任务状态更改为执行中

#### **6.3.3数据导入和导出测试**

（1）测试系统是否能正确读取和解析Excel文件中的数据

（2）测试系统是否能正确将数据导出为Excel文件格式

（3）测试数据导出的Excel文件是否能够正确打开并包含所有导出的数据

#### **6.3.4统计分析和图表分析测试**

（1）测试系统是否能正确对数据进行统计分析和图表分析

（2）测试系统生成的统计和图表是否准确反映了数据的特征和趋势

（3）测试系统是否能够按照用户选择的条件和时间范围生成相关的统计和图表

#### **6.3.5各模块功能测试**

（1）测试地图模块是否能正确展示管线，并能够正常响应用户的操作，包括编辑事件、审核事件等操作

（2）测试事件管理模块是否能正确显示所有事件的详细信息，并能够让管理员对事件进行处理、指派任务等操作

（3）测试人员管理模块是否能正确管理管道巡线人员和抢险人员的信息，并能够根据人员类型进行过滤和搜索

（4）测试任务中心模块是否能正确管理抢险任务，并能够根据现有失效事件新建任务并指派给相应的抢险人员

（5）测试信息管理模块是否能够正确管理各用户的个人信息，并能够包含用户头像、昵称、电话号码、邮箱等信息

### 6.4测试用例及结果

软件测试分白盒测试和黑盒测试两种，前者主要测试的是产品功能是否可以实现，不在乎产品内部的结构，只针对产品的功能做出检测评价；后者在测试之前必须对产品的内部结构进行熟悉和掌握，从而通过各种编程语言设计测试用例，实现软件的测试[[[4]](#endnote-3)]。除此以外，在设计测试用例，应根据软件测试的原则，选择那些发现错误可能性大的数据作为测试的输入数据。

下面将对本系统的各模块进行测试用例的列举和说明:

**表 6.4.1地图模块测试样例**

|  |  |  |
| --- | --- | --- |
| **测试数据** | **期望结果** | **实际结果** |
| 点击地图上的管线点、新建事件填写事件类型、位置、描述信息并提交 | 提交成功，事件上报到系统，可在事件统计页面查看 | 提交成功，事件上报到系统，可在事件统计页面查看 |
| 管理员点击报警事件 | 弹出事件审核框 | 弹出事件审核框 |
| 查看事件详细信息 | 显示事件的类型、位置、描述等信息 | 显示事件的类型、位置、描述等信息 |
| 修改事件状态 | 事件状态被修改，例如从“未处理”改为“处理中” | 事件状态修改成功 |
| 指派抢险任务 | 任务被指派到相应的抢险人员 | 任务被指派到相应的抢险人员 |
| 显示失效事件提示列表 | 根据紧急程度进行排序并提示失效事件 | 根据紧急程度进行排序并提示失效事件 |

**表 6.4.2事件管理模块测试样例**

|  |  |  |
| --- | --- | --- |
| **测试数据** | **期望结果** | **实际结果** |
| 根据事件状态、位置、时间等条件进行事件检索 | 根据条件筛选出符合要求的事件列表 | 根据条件筛选出符合要求的事件列表 |
| 新增事件 | 弹出新增事件表单、提交表单信息后，事件新增成功 | 弹出新增事件表单、提交表单信息后，事件新增成功 |
| 编辑事件状态和信息 | 事件状态被修改，事件信息被成功修改 | 事件状态被修改，事件信息被成功修改 |
| 删除事件 | 事件删除成功 | 事件删除成功 |
| 批量删除事件 | 批量删除事件成功 | 批量删除事件成功 |

**表 6.4.3人员管理模块测试样例**

|  |  |  |
| --- | --- | --- |
| **测试数据** | **期望结果** | **实际结果** |
| 新建管道巡线人员或抢险人员 | 人员信息被成功添加到系统中 | 人员信息被成功添加到系统中 |
| 编辑管道巡线人员或抢险人员的信息 | 人员信息被成功修改 | 人员信息被成功修改 |
| 删除管道巡线人员或抢险人员 | 人员信息被成功删除 | 人员信息被成功删除 |
| 根据人员类型进行过滤和搜索 | 根据人员类型筛选出符合要求的人员列表 | 根据人员类型筛选出符合要求的人员列表 |
| 批量删除人员 | 批量删除人员成功 | 批量删除人员成功 |

**表 6.4.4任务中心模块测试样例**

|  |  |  |
| --- | --- | --- |
| **测试数据** | **期望结果** | **实际结果** |
| 点击新建任务按钮 | 弹出新建任务对话框 | 弹出新建任务对话框 |
| 选择失效事件并指派抢险人员 | 任务状态变为“已指派” | 任务状态变为“已指派” |
| 对任务进行编辑并保存 | 任务信息更新成功 | 任务信息更新成功 |
| 点击删除按钮 | 任务成功删除 | 任务成功删除 |
| 根据任务状态进行过滤和搜索 | 显示符合条件的任务列表 | 显示符合条件的任务列表 |
| 根据任务负责人进行过滤和搜索 | 显示符合条件的任务列表 | 显示符合条件的任务列表 |

**表 6.4.5用户信息模块测试样例**

|  |  |  |
| --- | --- | --- |
| **测试数据** | **期望结果** | **实际结果** |
| 点击头像 | 头像放大 | 头像放大 |
| 修改昵称、电话号码、邮箱等信息 | 信息修改成功 | 信息修改成功 |
| 修改用户密码 | 密码修改成功 | 密码修改成功 |
| 退出登录 | 退回到登陆页面并且不能通过url访问其他页面 | 退回到登陆页面并且不能通过url访问其他页面 |

# 7 总结与展望

本文结合燃气管道失效抢险系统的项目结构与设计情况，参考了石油燃气企业正式使用的软件系统，结合实际的燃气管道抢险流程，从多角度阐述了本项目的开发过程。在需求分析阶段，从理论上模拟用户操作流程之后，再对其他的类似系统进行了比较研究，结合两者之后设计出本系统的需求。在开发环节中，首先对地图api的使用进行了细致的研究，在结合需求分析中设计的用例图，决定采用NodeJs+koa框架做后端开发，Vue+ElementUI+echarts做前端开发，Mysql数据库管理数据，ArcGis结合天地图来实现地图模块，保证了燃气管道失效抢险系统的技术成熟性，实现了系统的主要功能。本文所做的具体工作如下：

对燃气管道失效抢险系统的业务流程和使用功能进行了详细的分析，对失效事件相关的需求进行了需求分析，列出了系统要实现的预期目标和建设过程。

但是在完成该系统时，由于有些技术平时接触的比较少，在实际开发时，只是勉强完成预期功能，还存在很多未完善的地方。具体如下：

（1）在系统的可维护性上，做的努力还不够，燃气管道失效抢险系统应该是一个需要长期维护、进行功能扩展的软件系统，需要更多的对它的可扩展性、可移植性做支持。

（2）在对多用户的权限管理上还不够清晰，在多种不同类型的用户的权限控制上应该要秉持合理、严格、安全性等原则，同时，如何照顾各用户的使用体验也是需要更多的思考。

（3）在数据图表上应该为用户提供更多的图表类型和操作功能，提升系统的可视化能力。

# 致谢

王老师是一位非常有经验和才华的教授，他在我的研究方向上给了我很多的指导和建议。他的专业知识和丰富的经验让我在研究过程中得到了非常有价值的启示和帮助。在他的帮助下，我对自己的毕业课题有了更深刻的理解和认识。

在本次毕业设计过程中，我得到了王兵老师的无私帮助，在分配毕业课题之后，王老师认真仔细地为我分析了课题的主体功能框架和开发过程中可能遇到的难点，给了我很多启发，让我少走了很多弯路，设计的任务书时间规划也非常合理，让我在充裕的时间内，尽可能的去做好了毕业设计。

最后，我要感谢王老师为我的未来奠定了坚实的基础。通过王老师的教育和指导，我不仅获得了专业的知识和技能，还培养了独立思考、勇于创新和不断追求进步的精神。这些都将对我未来的学习和事业产生深远的影响。希望以后有机会能够向王老师请教学习，汲取更多的知识和经验。

1. []许康宁.TUBISTM系统[J].上海煤气,2002(03):9-10. [↑](#endnote-ref-0)
2. []王平,金辰捷,田贵云,彭国华.基于无线传感网络的远程燃气管道泄漏检测系统[J].无损检测,2012,34(12):25-28 [↑](#endnote-ref-1)
3. []王新颖,杨泰旺,宋兴帅,陈海群.卷积神经网络在燃气管道故障诊断中的应用[J].工业安全与环保,2019,45(02):36-40+68 [↑](#endnote-ref-2)
4. []妥泽花.基于黑盒测试与白盒测试的比较探究[J].电子世界,2021,No.617(11):55-56.DOI:10.19353/j.cnki.dzsj.2021.11.027. [↑](#endnote-ref-3)