



泗妖六

(



) 概要设计说明书



文件状态: 正式发布

文件标识:

当前版本: 1.0

作者:

姓名: 张石一、陈治中...

学号:

团队:

完成日期:

版本更新信息:

## 1 引言

### 1.1 使用人员:

张石一、陈治中、王伟杰、杨鹏飞、刘子昂、胡志国、孙博君、朱康晟

### 1.2 编写目的:

本阶段是在系统的需求分析的基础上,对地震灾情上报系统做的概要设计。主要解决实现该系统需求的模块设计问题。包括把该系统划分成哪些模块、决定各个模块之间的接口、模块之间传递的信息等。

在下一阶段的详细设计中,设计人员可参考此概要设计说明书,在概要设计对灾情上报系统所做的模块结构设计的基础上,对系统进行详细设计。在以后的软件测试以及软件维护阶段也可参考此说明书,以便于了解在概要设计过程中所完成的各模块设计结构,或在修改时找出在本阶段设计的不足或错误。

### 1.3 背景:

本项目将完成一个多源灾情上报系统。多源灾情数据通过接口输入到多源灾情数据管理服务系统平台,进行一体化编码;然后将编码后的数据输入到虚拟化管理系统。数据来源包括业务报送数据、泛在感知数据、舆情感知数据和承载体基础数据等,将来可以扩展更多数据来源。该系统依据数据的时效性进行数据存储,当前时刻为,设置合适的时间窗口,对内的数据进行存储,并且随着时间的延续,实现新数据存储旧数据淘汰;最后,当外界发出数据请求时,从管理系统中获取目标数据,并通过接口发送给请求方。需要设计一个通用的架构以适应众多异源异构的数据请求(例如多层架构中的每层低耦合,通用化接口 - 例如前后端分离,可以适用需求的变化)。实时处理技术的研究,实现灾情数据统一管理和高效合理利用,构建多源社会灾情数据管理服务系统。

### 1.4 定义与缩写

术语: SQL SERVER

解释:

系统服务器所使用的数据库管理系统(DBMS)。

术语: 事务流

解释:

数据进入模块后可能有多种路径进行处理。

术语: ROLLBACK

解释:

数据库的错误恢复机制

术语：系统

解释：

若未特别指出，特指本灾情上报系统

术语：SQL

解释：

Structured Query Language

术语：DBMS

解释：

Database Management System

## 1.5 参考资料

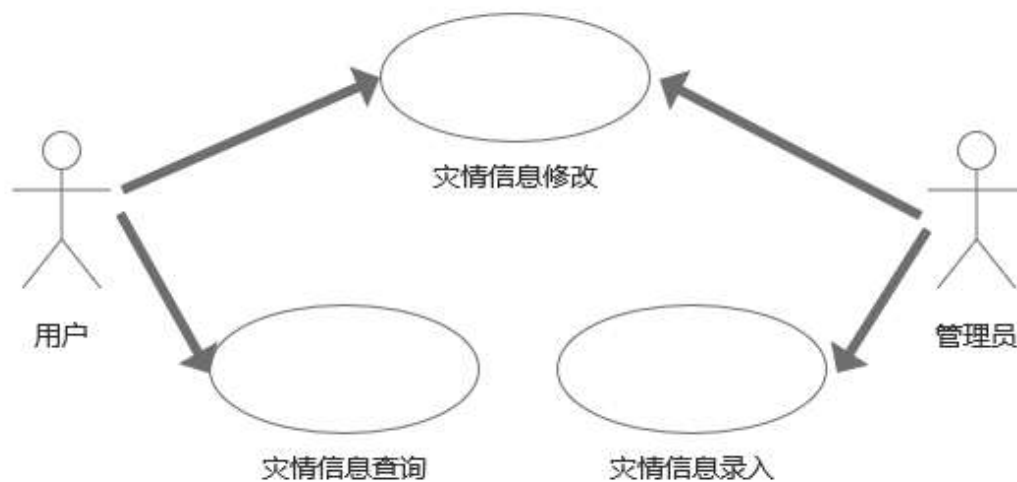
- [1] 《系统项目计划任务书》，大鹏航空公司，2005/7
- [2] 《系统项目开发计划》，软件工程科研组，2005/7
- [3] 《软件需求说明书》，软件工程科研组，2005/7
- [4] 《用户操作手册（初稿）》，软件工程科研组，2005/4

## 2 总体设计

### 2.1 需求规定：

- 对功能的规定

本系统用于灾情信息查询统计，包括信息录入，信息查询，信息修改等。基本功能划分如下图



各模块具体解释及含义：

用例名称：灾情信息查询

用例描述：当用户查询灾情信息时，这个用例开始；它对用户查询的信息做出处理；当用户停止查询时结束

活动者：用户

位置：用户的电脑

状态：已定义路径

假设：查询值无论为真为假都可以随时退出，执行其他模块功能

先决条件：用户要先登录到网站主页面



后续条件：退出  
 主路径：用户根据具体需要查询信息，包括灾情发生时间地点，具体损失信息  
 例外路径：用户进行查询，但该航班信息不存在  
 用例名称：灾情信息修改  
 用例描述：该用例从用户注册登录网站开始，对信息问题进行反馈，在管理员修改后该用例结束  
 活动者：用户和管理员  
 位置：用户的电脑和管理员的电脑  
 状态：已定义路径  
 假设：当用户对信息进行反馈后，管理员才能进行修改  
 先决条件：用户要先注册登录  
 后决条件：用户注销；管理员对信息进行修改  
 主路径：用户对信息问题进行反馈  
 例外路径：用户进行问题反馈，但反馈未在管理员处通过  
 用例名称：灾情信息录入  
 用例描述：该用例从管理员处理灾情信息开始，将信息录入系统，在管理员操作完毕时该用例结束  
 活动者：管理员  
 位置：管理员的电脑  
 状态：已定义路径  
 假设：获得管理员权限才能执行操作  
 先决条件：管理员要先登录到管理系统  
 后决条件：退出管理系统  
 主路径：管理员进行灾情信息的录入  
 例外路径：管理员对灾情信息的录入不成功  
 • 对性能的规定  
 精度：在从数据库中提取数据时，要求数据定位准确；在向数据库输入数据时，要求录入数据有校验精确；查询返回的信息均来源于数据库  
 时间特性要求：查询响应时间应在人的感官视觉时间范围内；更新处理、数据传输和运行时间受到网络个人操作设备的影响会对具体的操作有不同的反应，且随着系统的不断更新、版本的升级（在硬件设备允许的条件下）会有所提高  
 • 输入输出要求

#### 输入数据

数据项名称	说明	类型	长度
时间	用户从页面中自主选择想要查找的灾情的时间，可用于查询具体灾情信息	日期型	20字节
灾情属性	用户从页面中自主选择想要查找的灾情的属性（泥石流、地震、火山爆发等），可用于查询具体灾情信息	字符串型	10字节
经度	用户从页面中自主选择想要查找的灾情的经度，可用于查询具体灾情信息	浮点型	10字节
纬度	用户从页面中自主选择想要查找的灾情的纬度，可用于查询具体灾情信息	浮点型	10字节

#### 输出数据

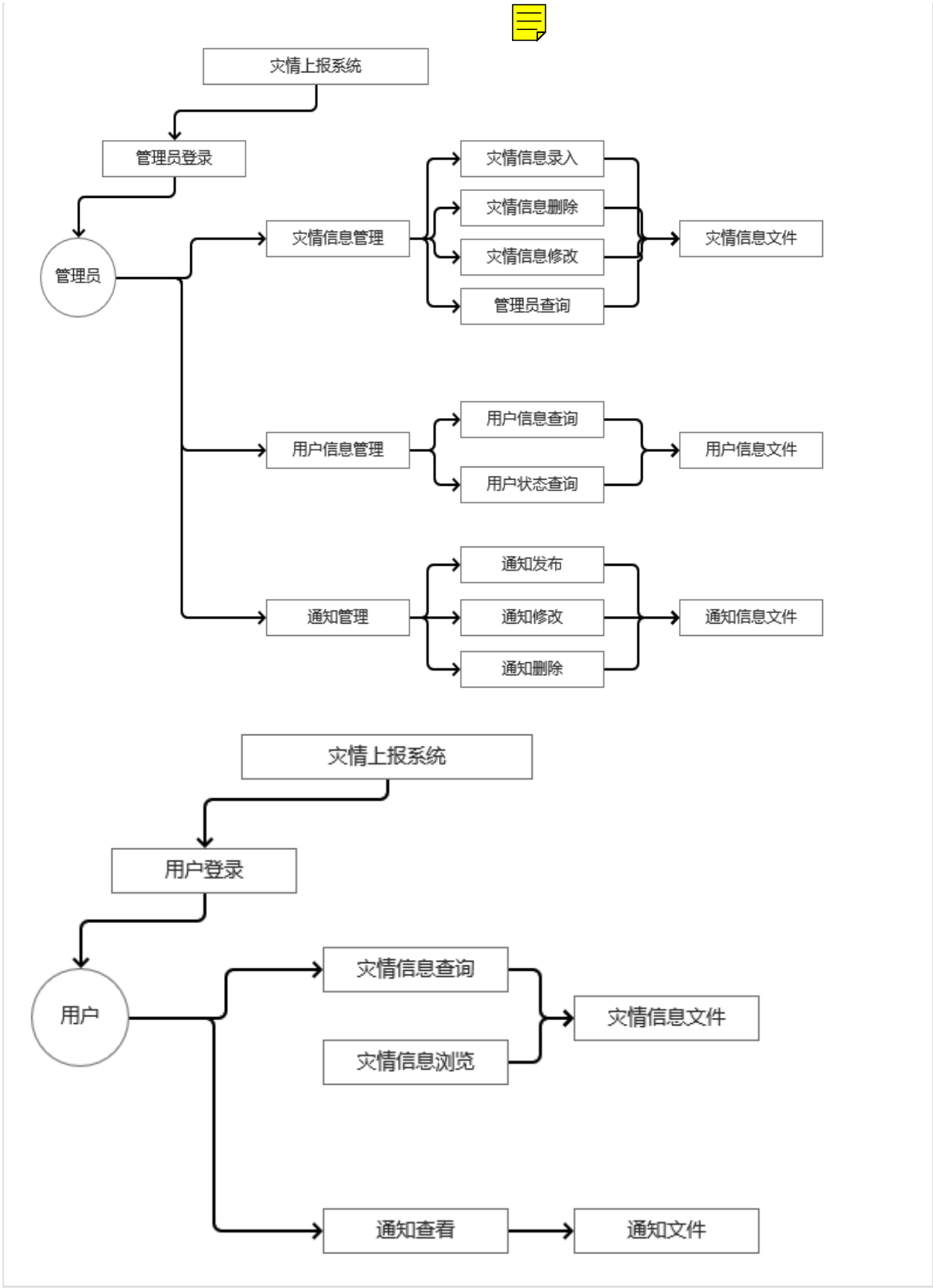
数据项名称	说明	类型	长度
时间	用户查询信息后的返回值	日期型	20字节
灾情属性	用户查询信息后的返回值	字符串型	10字节
经度	用户查询信息后的返回值	浮点型	10字节
纬度	用户查询信息后的返回值	浮点型	10字节

## 2.2 运行环境:

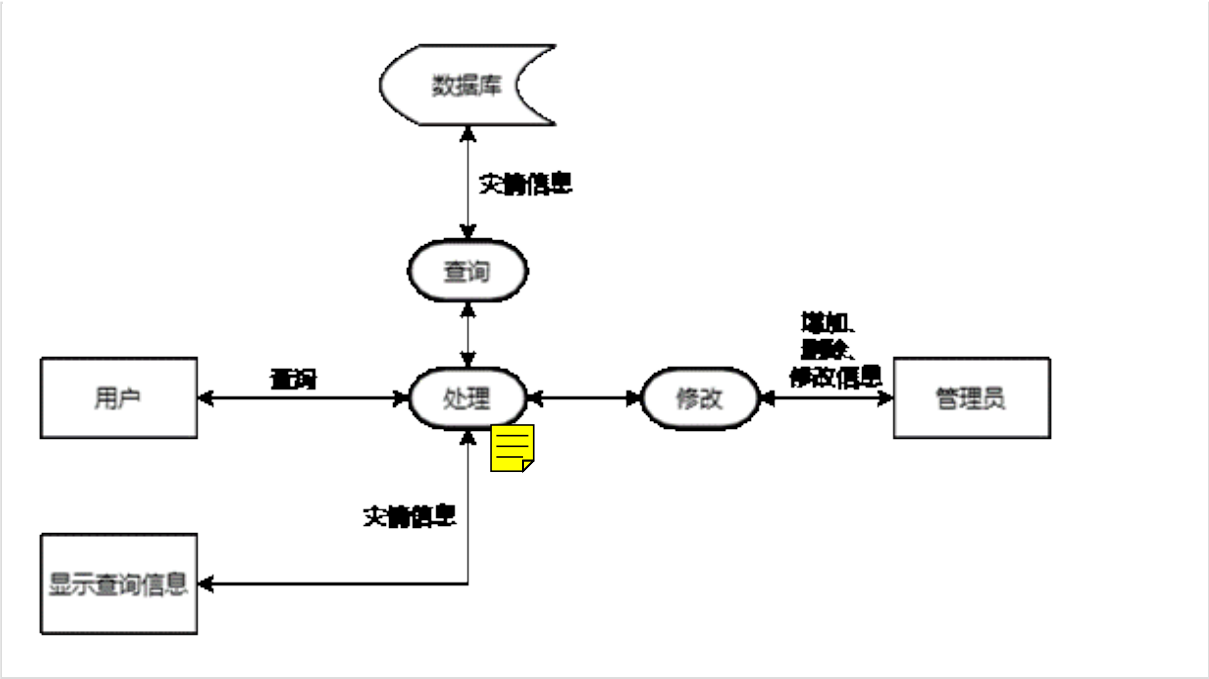
推荐使用：Windows 10，数据库MySQL 5.7  
 计算机配置要求：内存4G以上，硬盘40G以上，能够连接网络

## 2.3 基本设计概念和处理流程:

针对本系统的管理员和用户两种不同的角色，系统的处理流程如下图所示：



2.4 结构:



2.5 功能需求与个部分程序的关系：

	增添数据	删除数据	修改数据	查询数据	
用户查询				#	
更新信息	#	#	#		

2.6 人工处理过程：

人工操作为管理员 更新灾情数据，包括增加、删除、修改。

2.7 尚未问决的问题：

暂无

2.8 自由表述：

3 接口设计

3.1 用户接口：

signUp(string name ,string password) bool state  
注册接口 用户输入用户名和密码，前端判断字符串长度在后端与数据库中信息对比是不是有重复，后端返回状态信息，根据状态信息判断是否可以登录。  
login(string name ,string password) bool state  
登录接口 用户输入用户名和密码，前端判断字符串长度在后端与数据库中信息对比信息，后端返回状态信息，根据状态信息判断是否可以登录。

```
getMap() structure[] object  
获取地图模式的接口，发送获取地图请求到后端，后端返回灾情结构体数组。  
select(structure object) structure[] object  
根据条件查询的接口 并返回所查到符合条件的信息
```

### 3.2 外部接口

硬件接口：

无

软件接口：

高德地图 API

### 3.3 内部接口

硬件接口：

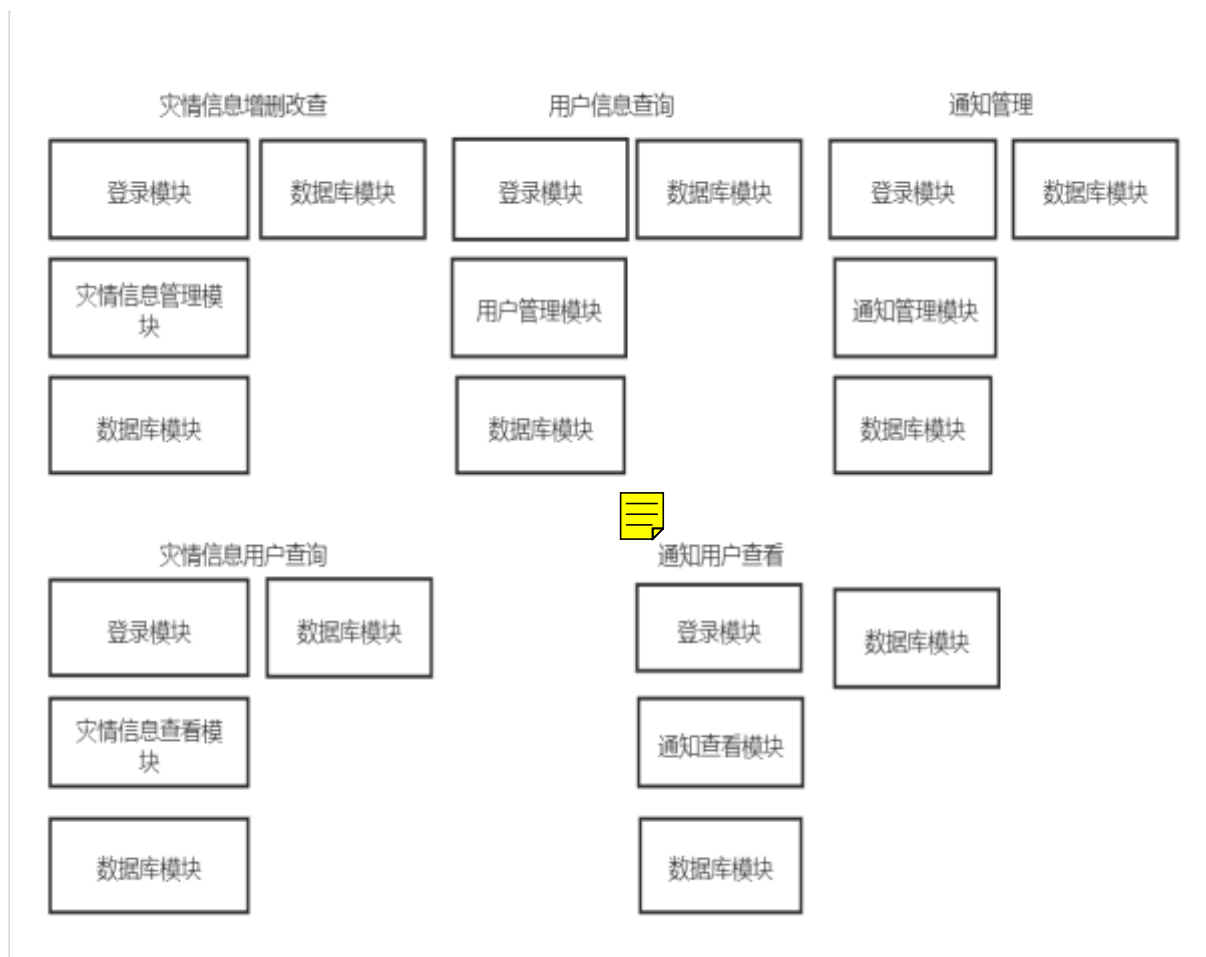
无

软件接口：

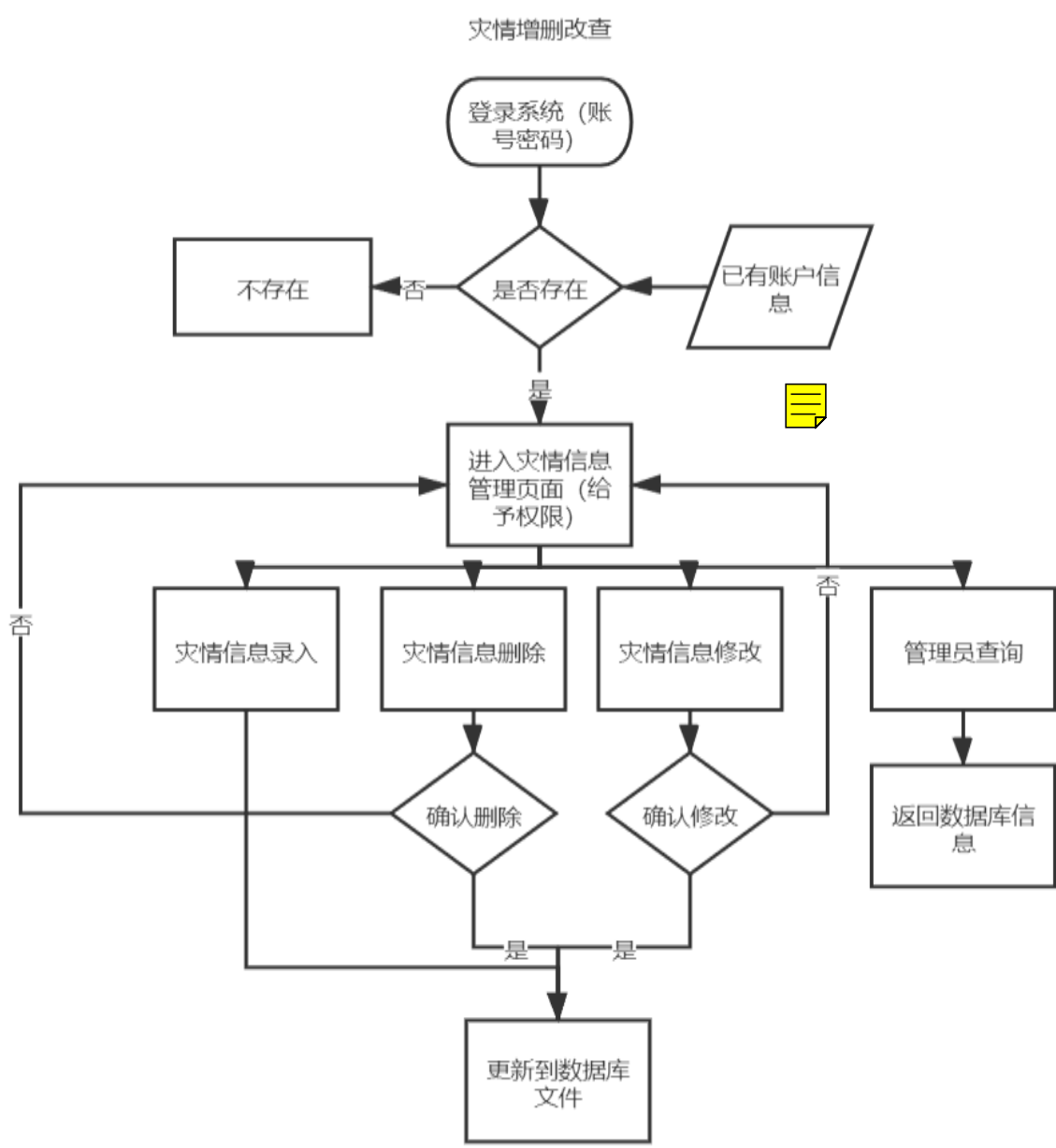
```
login(string name ,string password) bool state  
登录接口 用户输入用户名和密码，前端判断字符串长度在后端与数据库中信息对比信息，后端返回状态信息，根据状态信息判断是否可以登录。  
addDisaster(structure object, object) bool  
管理员的灾情添加接口 第一个结构体为用户结构体用来验证身份保证安全，第二个结构体为灾情结构体
```

## 4 运行设计

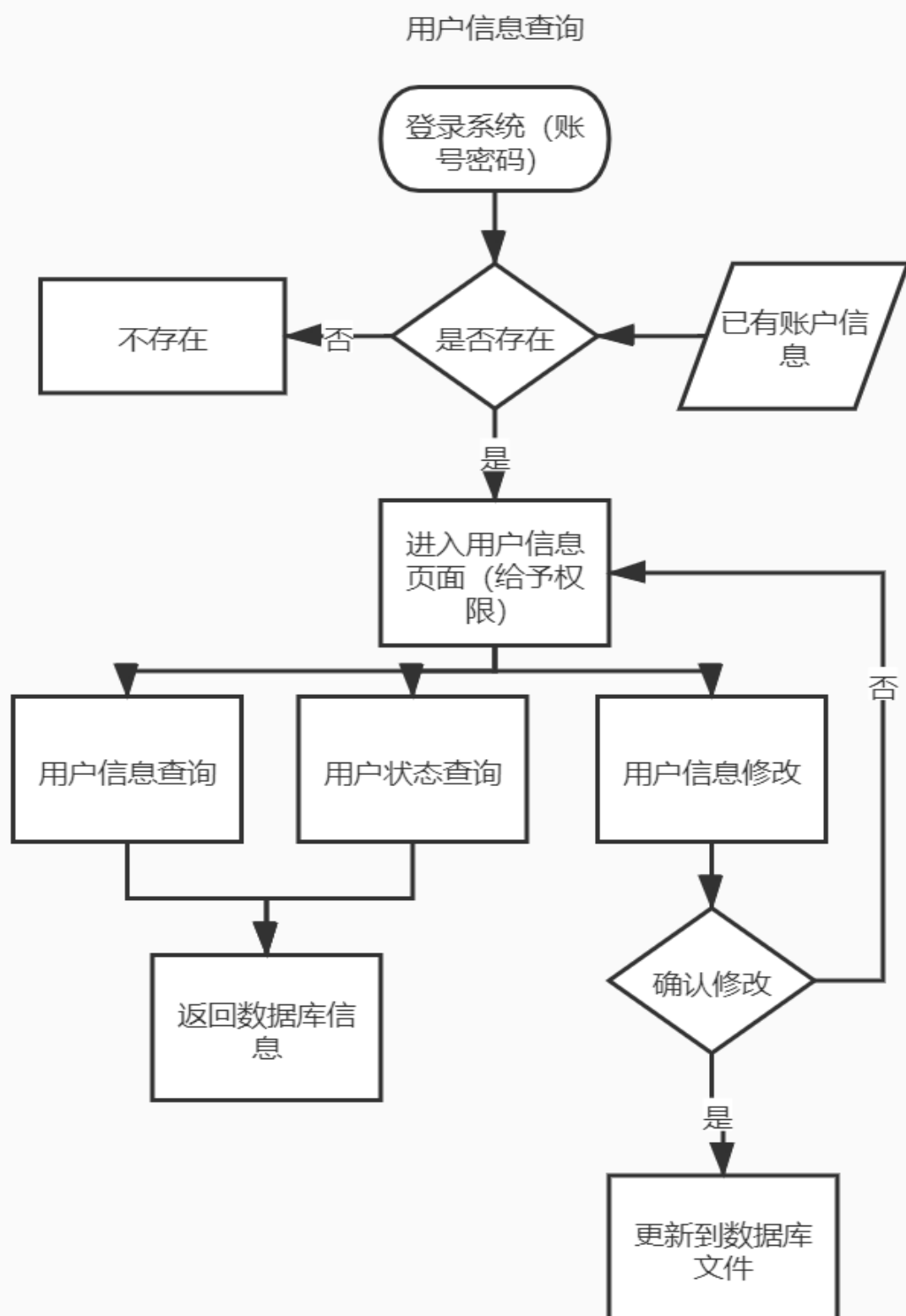
### 4.1 运行模块组合：

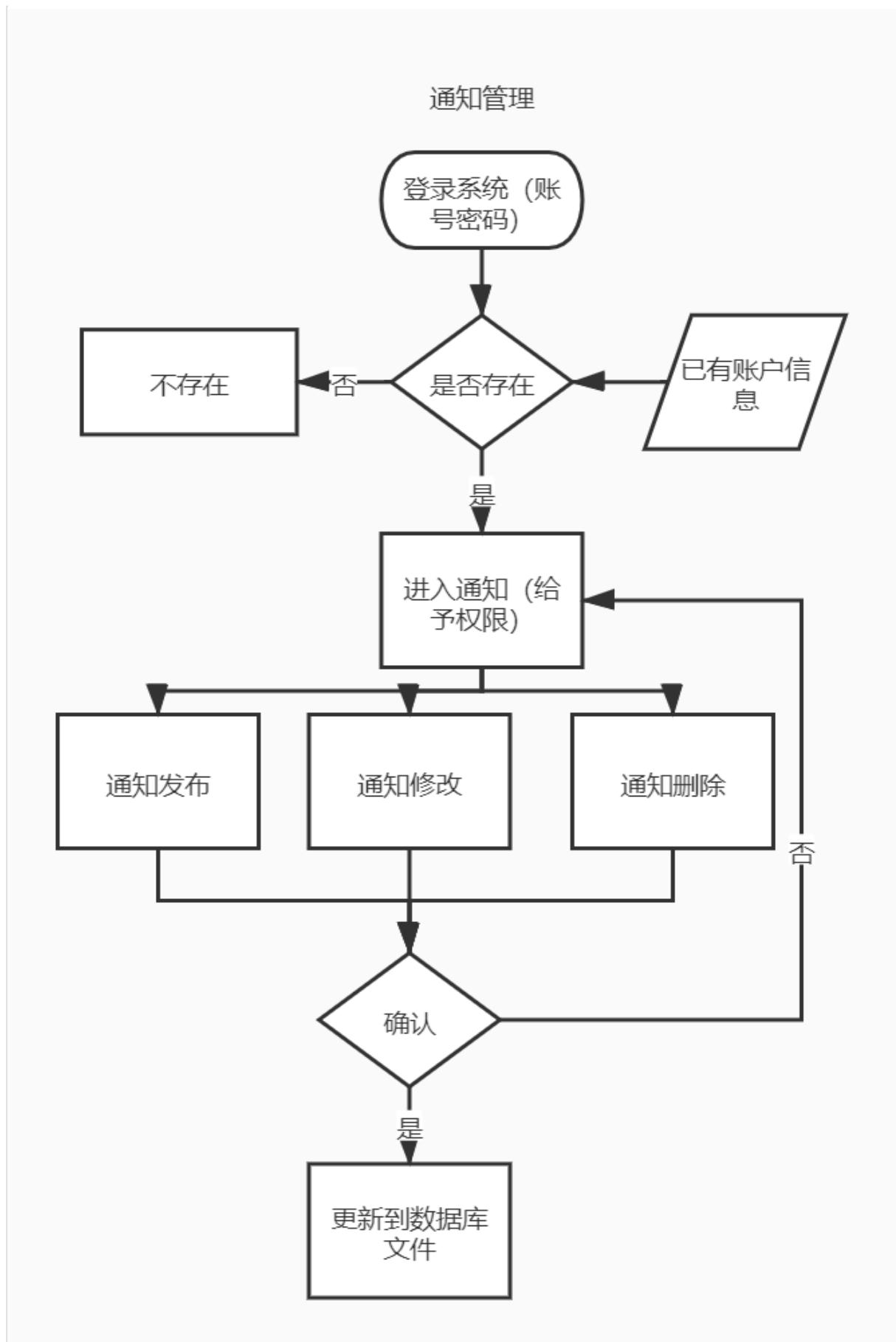


#### 4.2 运行控制:

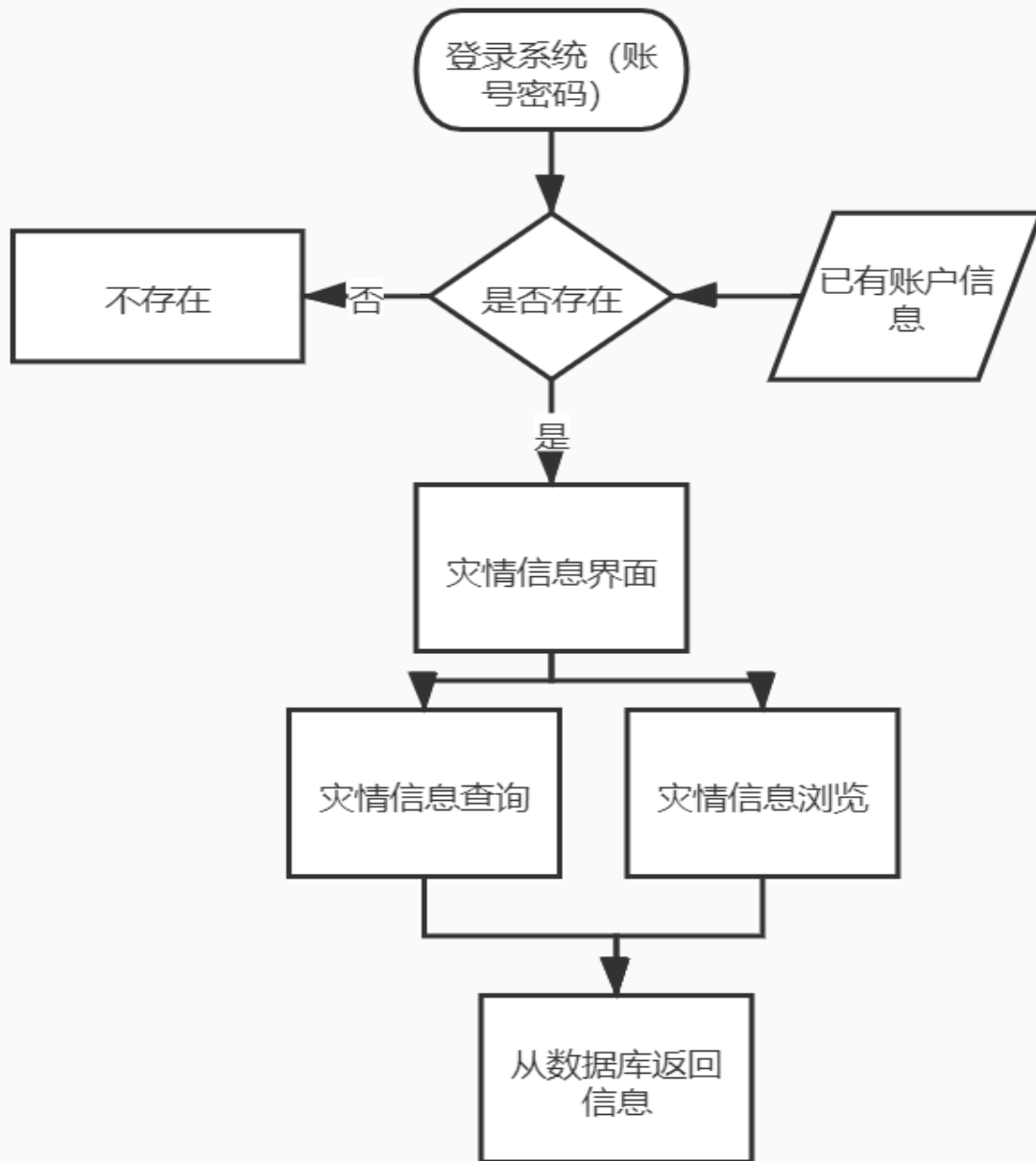


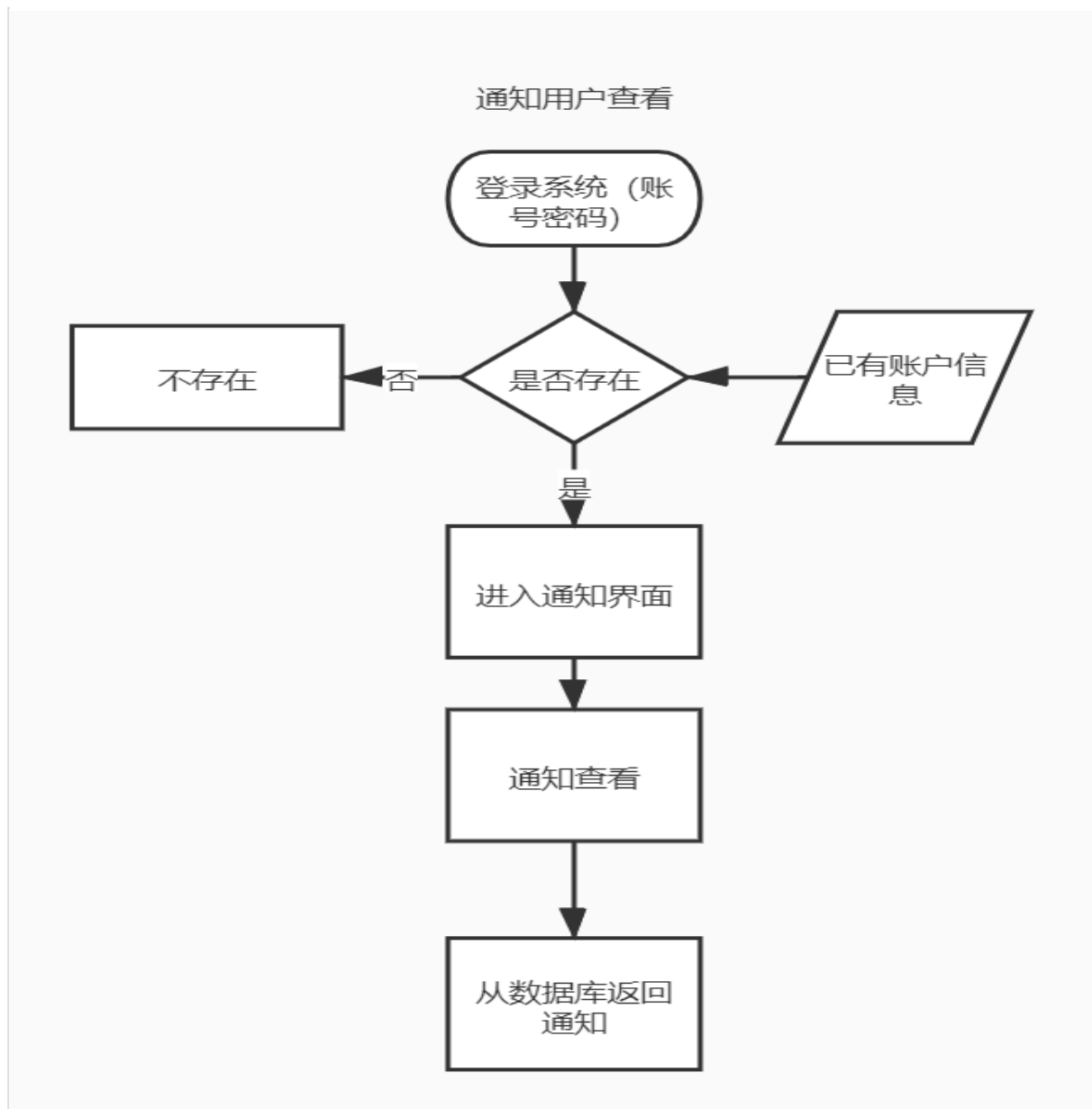






### 灾情信息用户查询





#### 4.3 运行时间:

灾情信息增删改: 对灾害信息表占用不超过100ms  
 用户查询: 当检测到需要对文件进行更改, 对用户信息表占用不超过49ms  
 通知查询: 当检测到需要对通知文件进行更新, 对通知文件占用不超过50ms  
 灾情信息用户查询: 当检测到需要对文件进行更新, 对灾情信息表占用不超过58ms  
 通知用户查看: 当检测到通知文件进行更新, 对通知文件占用不超过57ms

#### 4.4 自由表述:

### 5 系统数据结构设计

#### 5.1 逻辑结构设计要点:

```

naturalDisaster{string time, string attribute, double longitude , latitude,
}
user{string name, string username, password, bool identity,
}
  
```

## 5.2 物理结构设计要点:

我们设计的重点是时间和空间效率

(1) 确定数据的存储结构

灾情上报系统, 将会在后台进行数据处理成xml或json格式后, 进行数据信息的上传。确定数据库存储结构时要综合考虑存取时间、存储空间利用率和维护代价三方面的因素。这三个方面常常是相互矛盾的, 例如消除一切冗余数据虽然能够节约存储空间, 但往往会导致检索代价的增加, 因此必须进行权衡, 选择一个折中方案。

(2) 设计数据的存取路径

建立索引, 在数据库后台我们会进行实时更新。对于DML效率的要求, 我们用SQL Server中默认使用索引组织表来完成。表分割方面使用本地分区视图方式进行表分区。

(3) 确定数据的存放位置

为了提高系统性能, 灾情信息数据应该根据应用情况将易变部分与稳定部分、经常存取部分和存取频率较低部分分开存放。

(4) 确定系统配置

DBMS产品一般都提供

## 5.3 数据结构与程序的关系:

数据结构与程序是软件的重要组成部分, 程序的正确执行依赖于合理的数据结构。

系统当在用户进行灾情上报时需要数据库数据结构进行查询和修改; 在上报过程中需要对数据库中的所有表, 进行联合查询, 修改。

## 5.4 自由表述:

# 6 系统出错处理设计

## 6.1 错误处理

出错信息:

程序在运行时主要会出现两种错误:

1、由于输入信息, 或无法满足要求时产生的错误, 称为软错误。

2、由于其他问题, 如网络传输超时等, 产生的问题, 称为硬错误。

对于软错误, 须在输入数据验证模块由数据进行数据分析, 判断错误类型, 再生成相应的错误提示语句, 送到输出模块中。

对于硬错误, 可在出错的相应模块中输出简单的出错语句, 并将程序重置。返回输入阶段。

列如

1. 用户名和密码错误, 提示输入不正确, 返回重新输入。

2. 验证码输入错误, 提示验证码错误, 返回重新输入。

3. 账号不存在, 提示账号不存在, 是否注册。

4. 注册密码太简单, 提示安全性低, 重新输入。

5. 输入格式错误, 提示输入格式不对, 应该输入什么格式。

6. 新密码与确认密码不匹配, 提示密码不匹配, 请重新输入。

7. 信息类型大于50个字符, 提示超过规定字数, 重新输入。

8. 输入信息为空, 提示未输入, 请输入。

9. 不合法的URL请求, 提示错误信息, 返回登陆界面。

10. 程序出错抛出异常, 输入log日志中的错误信息。

11. 数据量访问过大, 返回请求失败信息。

12. 登陆调用web服务器方法, 返回登陆页面, 要求用户登陆

补救措施:

1. 后备技术说明准备采用的后备技术, 当原始系统数据万一丢失时启用的副本的建立和启动的技术, 例如周期性地把磁盘信息记录到磁带上就是对于磁盘媒体的一种后备技术;

2. 降效技术说明准备采用的后备技术, 使用另一个效率稍低的系统或方法来求得所需结果的某些部分, 例如一个自动系统的降效技术可以是手工操作和数据的人工记录;

3. 恢复及再启动技术说明将使用的恢复再启动技术, 使软件从故障点恢复执

4. 系统在用户登录后台管理平台失败时提示“账户信息不正确”。

5. 用户在选择自己没有权限使用的模块时提示“抱歉, 你没有改模块的管理权限”。

6. 用户保存数据失败时提示“保存失败”。
7. 用户在删除重要信息时弹出警告框提示用户是否确认要删除数据。
8. 用户在无法访问页面时显示“该页面找不到了”或者“页面维护中”。
9. 数据在数据库中已经有备份，所以系统出错后可以依靠数据库的恢复功能，并且依靠日志文件使系统再启动，就算系统崩溃用户数据也不会丢失或遭到破坏。但有可能占用更多的数据存储空间，权衡措施由用户来决定。
10. 开始时间与结束时间比较，开始时间比结束时间要大，应提示：起始时间不能大于结束时间。

## 6.2 系统维护设计：

### 测试点

#### 一、输入框

##### 1、字符型输入框：

(1) 字符型输入框：中文，英文全角、英文半角、数字、空或者空格或者回车、特殊字符（~!@#¥%……&\*? [] {} ”（特别要注意单引号和&符号））。禁止直接输入特殊字符时，使用“复制+粘贴”功能尝试输入。

(2) 长度检查：最小长度、最大长度、最小长度-1、最大长度+1、输入超长字符比如把整个文章拷贝过去。

(3) 空格检查：输入的字符间有空格、字符前有空格、字符后有空格、字符前后有空格

(4) 多行文本框输入：允许回车换行、保存后再显示能够保存输入的格式、仅输入回车换行，检查能否正确保存（若能，检查保存结果，若不能，查看是否有正常提示）

(5) 安全性检查：输入特殊字符串

（null, NULL, , javascript, <script>, </script>, <title></title>, <html></html>, <td></td>）、输入脚本函数（<script>alert("abc")</script>）、document.write("abc")、<b>hello</b>、sql注入）

##### 2、数值型输入框：

(1) 边界值：最大值、最小值、最大值+1、最小值-1

(2) 位数：最小位数、最大位数、最小位数-1、最大位数+1、输入超长值

(3) 特殊字符：输入空白（NULL）、空格或“~!@#%&\*() +{} |[]\:"'<>?;’,./?:;-”等可能导致系统错误的字符、禁止直接输入特殊字符时，尝试使用粘贴拷贝查看是否能正常提交、word中的特殊功能，通过剪贴板拷贝到输入框，分页符，分节符类似公式的上下标等、数值的特殊符号如Σ，log，ln，Π，+，-等

(4) 异常值：输入负整数、负小数、分数、输入字母或汉字、小数（小数前0点舍去的情况，多个小数点的情况）、首位为0的数字如01、02、科学计数法是否支持1.0E2、全角数字与半角数字、数字与字母混合、16进制，8进制数值、货币型输入（允许小数点后面几位）

(5) 安全性检查：不能直接输入就copy，输入内容如上

##### 3、日期型输入框：

(1) 合法性检查：（输入0日、1日、32日）、月输入[1、3、5、7、8、10、12]、日输入[31]、月输入[4、6、9、11]、日输入[30][31]、输入非闰年，月输入[2]，日期输入[28、29]、输入闰年，月输入[2]、日期输入[29、30]、月输入[0、1、12、13]

(2) 异常值、特殊字符：输入空白或NULL、输入“~!@#¥%……&\*（）{} []”等可能导致系统错误的字符

(3) 安全性检查：不能直接输入，就copy，是否数据检验出错

4、信息重复：在一些需要命名，且名字应该唯一的信息输入重复的名字或ID，看系统有没有处理，会否报错，重名包括是否区分大小写，以及在输入内容的前后输入空格，系统是否作出正确处理

#### 二、搜索功能

若查询条件为输入框，则参考输入框对应类型的测试方法

##### 1、功能实现：

(1) 如果支持模糊查询，搜索名称中任意一个字符是否能搜索到

(2) 比较长的名称是否能查到

(3) 输入系统中不存在的与之匹配的条件

(4) 用户进行查询操作时，一般情况是不进行查询条件的清空，除非需求特殊说明。

(5) 拼音查询

##### 2、组合测试：

(1) 不同查询条件之间来回选择，是否出现页面错误（单选框和多选框最容易出错）

(2) 测试多个查询条件时，要注意查询条件的组合测试，可能不同组合的测试会报错。

#### 三、添加、修改功能

1、特殊键：（1）是否支持Tab键 （2）是否支持回车键

2、提示信息：（1）不符合要求的地方是否有错误提示

3、唯一性：（1）字段唯一的，是否可以重复添加，添加后是否能修改为已存在的字段（字段包括区分大小写以及在输入的内容前后输入空格，保存后，数据是否真的插入到数据库中，注意保存后数据的正确性）

##### 4、数据正确性：

(1) 对编辑页的每个编辑项进行修改，点击保存，是否可以保存成功，检查想关联的数据是否得到更新。

(2) 进行必填项检查（即是否给出提示以及提示后是否依然把数据存到数据库中；是否提示后出现页码错乱等）

(3) 是否能够连续添加（针对特殊情况）

(4) 在编辑的时候，注意编辑项的长度限制，有时在添加的时候有，在编辑的时候却没有（注意要添加和修改规则是否一致）

(5) 对于有图片上传功能的编辑框，若不上传图片，查看编辑页面时是否显示有默认的图片，若上传图

片，查看是否显示为上传图片

(6) 修改后增加数据后，特别要注意查询页面的数据是否及时更新，特别是在首页时要注意数据的更新。

(7) 提交数据时，连续多次点击，查看系统会不会连续增加几条相同的数据或报错。

(8) 若结果列表中没有记录或者没选择某条记录，点击修改按钮，系统会抛异常。

#### 四、删除功能

1、特殊键：(1) 是否支持Tab键 (2) 是否支持回车键

2、提示信息：(1) 不选择任何信息，直接点击删除按钮，是否有提示 (2) 删除某条信息时，应该有确认提示

3、数据 实现：(1) 是否能连续删除多个产品 (2) 当只有一条数据时，是否可以删除成功 (3) 删除一条数据后，是否可以添加相同的数据 (4) 如系统支持批量删除，注意删除的信息是否正确 (5) 如有全选，注意是否把所有的数据删除 (6) 删除数据时，要注意相应查询页面的数据是否及时更新 (7) 如删除的数据与其他业务数据关联，要注意其关联性 (如删除部门信息时，部门下游员工，则应该给出提示) (8) 如果结果列表中没有记录或没有选择任何一条记录，点击删除按钮系统会报错。

如：某一功能模块具有最基本的增删改查功能，则需要进行以下测试

单项功能测试 (增加、修改、查询、删除)

增加——>增加——>增加 (连续增加测试)

增加——>删除

增加——>删除——>增加 (新增加的内容与删除内容一致)

增加——>修改——>删除

修改——>修改——>修改 (连续修改测试)

修改——>增加 (新增加的内容与修改前内容一致)

修改——>删除

修改——>删除——>增加 (新增加的内容与删除内容一致)

删除——>删除——>删除 (连续删除测试)

#### 五、注册、登陆模块

1、注册功能：

(1) 注册时，设置密码为特殊版本号，检查登录时是否会报错

(2) 注册成功后，页面应该以登陆状态跳转到首页或指定页面

(3) 在注册成功后，删除注册的账号，检查是否可以注册成功

(4) 注册已注册的账号，检查是否可以注册成功

(5) 改变存在的用户的用户名和密码的大小写，来注册。(有的需求是区分大小写，有的不区分)

(6) 看是否支持tap和enter键等；密码是否可以复制粘贴；密码是否以\* 之类的加秘符号显示

(7) 对于账号输入与密码输入运用等价类、边界值写就好了，这里不再累述

2、登陆功能：

(1) 输入正确的用户名和正确的密码

(2) 输入正确的用户名和错误的密码

(3) 输入错误的用户名和正确的密码

(4) 输入错误的用户名和错误的密码

(5) 不输入用户名和密码 (均为空格)

(6) 只输入用户名，密码为空

(7) 用户名为空，只输入密码

(8) 输入正确的用户名和密码，但是不区分大小写

(9) 用户名和密码包括特殊字符

(10) 用户名和密码输入超长值

(11) 已删除的用户名和密码

(12) 登录时，当页面刷新或重新输入数据时，验证码是否更新

#### 六、上传图片

(1) 上传各种主流的图片格式文件

(2) 上传非图片格式的文件

(3) 上传图片文件的大小，几k，上百k，到几兆，几十兆

(4) 上传正在被其他应用使用的图片文件

(5) 上传的手动输入上传地址

(6) 上传地址输入不存在的图片地址

(7) 上传地址输入图片名称来上传

(8) 不选择文件直接点击上传，查看是否给出提示

(9) 连续多次选择不同的文件，查看是否上传最后一次选择的文件

(10) 同时上传多张图片文件

#### 七、查询结果列表

(1) 列表、列宽是否合理

(2) 列表数据太宽有没有提供横向滚动

(3) 列表的列名有没有与内容对应

(4) 列表的每列的列名是否描述的清晰

(5) 列表是否把不必要的列都显示出来

(6) 点击某列进行排序，是否会报错 (点击查看每一页的排序是否正确)

(7) 双击或单击某列信息，是否会报错

#### 维护数据字典

1. 使用第三方工具对数据字典进行维护

2. 利用数据库本身的备注字段来维护数据字典。
3. 导出数据字典

### 选择合适的列建立索引

1. 出现在where从句, group by 从句, order by 从句中的列
2. 可选择性高的列要放到索引的前面
3. 索引中不要包括太长的数据类型

### 注意事项

1. 索引并不是越多越好, 过多的索引不但会降低写效率, 而且会降低读的效率
2. 定期维护索引碎片
3. 在SQL语句中不要使用强制索引关键字

### 检查点:

#### 表命名和字段命名

表命名和字段命名要注意地方是:

- 一、命名不要使用中文, 因为中文的编码是GB2312或者GBK, 但数据库的默认编码都是UTF-8。
- 二、命名不要和系统函数或关键字重名, 因为写SQL语句的时候通常都不会加表示列名或者字段名的转义符号, 而如果表命名或者字段命名与系统函数或关键字重名, 就会出现SQL运行的错误, 而又要费时间去找, 找完还要把转义符都加上, 非常麻烦。
- 三、命名不要用拼音首字母, 使用英文或者拼音全拼, 因为这会使得别人很难读得懂字段的意思。

#### 字段类型定义

字段类型定义需要注意的地方是:

- 一、字段长度设置的长度稍微大一些, 特别是在定义varchar字符串类型的时候, 最好是估计长度的100%~120%, 因为在实际开发过程中, 一定会遇到需求不断变更, 或者出乎意料之外的情况, 如果在此之前做过一些准备, 那么即使出现意外了, 也能非常轻松的应对了。但不能设置太大, 比如像varchar这种, 虽然是可变长度的字符串, 但是如果设置太大, 是会影响查询性能的。
- 二、合理使用固定长度字符串类型, 固定长度字符串类型与可变长度字符串类型在查询方面具有较大优势, 固定长度字符串类型做索引性能要比可变长度字符串类型来的高很多, 具体是因为哈希值的特点。

#### 表结构设计

关于表结构的设计:

- 一、主键, 每新建一张表都要有主键, 主键是一个标识字段, 新建主键的时候要建立聚集索引, 聚集索引能提高查询性能, 这样设置有利于让各种数据库框架认识哪些字段是作为标识字段使用的。
- 二、表关系, 主要说明一对一关系:
  - 一对一关系, 单个对单个, 一对一关系更多的是用于做表扩展字段的, 这样的好处是能重用, 提高了扩展性。若直接加上这些字段, 当用户数量是几万个, 几十万个, 甚至几百万个, 数据冗余量就非常大了, 会非常影响数据库性能。
- 三、无法确定字段的处理, 当要记录参数时, 但又无法确定参数有哪些, 那么这个时候我们先确定一些最基本的参数, 若参数存在查询需求的, 则列为基本参数; 确定完基本参数后, 再加字段来分别用来确定存储结构, 和存储序列化数据的。

#### 数据冗余

当数据冗余对数据库性能影响不大, 或者性能要求不高, 它是可以存在的。比如权限, CMS权限有分为模型权限和栏目权限, 模型权限是以模型的增、删、改为一个单元的, 栏目权限是以栏目是否能访问为一个单元的, 这两个是权限里面内容字段是都需要被查询的, 那么直接把模型权限和栏目权限放在一张表里, 模型权限和栏目权限里面内容字段都是可空的, 但因为权限的数据量本身不会很大, 这些冗余量对数据库性能影响微乎其微。

#### 数据维护模块:

面对大量的数据信息, 管理系统至少应具备以下方面的功能:

将数据库中数据备份到指定驱动器软盘上, 以便保存原始数据、必要时可恢复到数据库中重新使用;

删除数据库中不必要或过期数据以减少磁盘空间损失, 同时也可提高对数据库操作时的效率, 加快系统的运行速度;

将备份到软盘上的数据恢复到系统的数据库中, 以便系统对某些资料重新计算或进行年终综合统计等操作。

针对以上几个方面设计通用的数据维护模块, 有效实现对数据库管理系统中各类数据的备份、删除、恢复等功能。

#### 数据备份:

将数据库中数据备份到软盘上之前, 在软盘上建立一个合适的、对应于该数据库的数据文件名, 这个文件名要体现出数据库库名的信息以及备份数据的日期信息, 日期信息以月为单位。使得备份到软盘上的文件简明易查, 又能有效地将数据文件恢复到系统对应数据库。我们对数据库名以及软盘上的数据文件名提出一些约定。例如, 某数据库名为HY48, 备份数据期为21年12月的数据, 我们希望数据文件名为HY2112.048。从这个数据文件名中, 我们规定前两位字母体现了数据库的一般信息-后四位数字为日期信息, 扩展名为数据库的编号信息。之后, 使用FOXBASE语言的标准数据SDF格式即可在软盘上生成数据文件。

#### 数据删除:

存入数据库中的数据会在计算、查询或打印报表时使用, 这些数据中有部分为系统的可变参数或不变参数, 管理系统在计算时将不断对它们进行读取。然而, 大部分数据在经过一段时间的使用以后短时期内不再使用。若这些数据长时间地存储在硬盘上不仅会占用有限的磁盘空间, 而且还会大大影响系统对数据库的操作速度。为提高系统运行速度以及减少磁盘空间的损失, 将数据备份到软盘上后, 从数据库中删除它们,



当新的月份开始时,过去月份的数据除少数需在年终统计时再调用外,大部分数据已完成其历史使命,因此管理系统会提供删除数据的机会,同时系统操作人员也会按时删除数据库内这部分数据。

**数据恢复:**

由于某些特殊原因,管理系统需要重新计算过去的的数据资料.系统需对全年的数据进行年终综合。此时用户有必要将已备份到软盘上的数据文件恢复到管理系统的数据库内。

为了实现在某数据日期段已备份的软盘数据文件恢复到系统对应的数据库中,程序会构造一个与备份数据时一致的数据文件名,然后与软盘上的数据文件名进行比较,当软盘上存作该数据文件时,即可将它恢复到对应的数据库内。数据文件名的产生完全相同于数据备份时产生数据文件名的方法。此时,使用FOXBASE提供的APPEND命令,以及标准数据SDF格式将软盘上的数据文件写入系统相应的数据库