**灾后寻人系统设计**

项目负责人 ：吕东林

作 者：吕东林  
 日 期：2016.4.13

**天脉拓道（北京）科技有限公司**

**修改记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **修改内容** | **修改人** | **日期** | **备注** |
| V1.0 | 完成总体设计 | 吕东林 | 2016-4-13 |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目 录**

[背景](#h.30j0zll)

[实现的功能](#h.2et92p0)

[设计考虑](#h.c6pao4hhb53t)

3.1 设计模式

3.2 运行环境

3.3业务流程/结构图

3.4系统架构图

3.5 语言选择

3.6 WEB框架

3.7 API设计规范

3.8 协议

3.10 安全性

详细设计

4.1数据表设计（仅列出主要数据表）

前端部份整体设计

5.1 用户部份UI

5.2 用户的操作流程

5.3 前端主要接口及说明

# 背景

互联网是新媒体的载体，网络传播有以下的几点优势：

1）以信息技术为基础的高速数据传递。

2）传播的快捷性，在发布的瞬间可以传递给受众，在内容上具有极强的时效性。

3）信息的大容量性，互联网络实现了在线资源共享，任何资料库内的信息资源只要联网，都成为公众的共享资源。

4）检索的便利性。利用搜索引擎或新闻站点等多种检索方式，可以快速地获得自己所需的信息。由于互联网本身的信息传播优势，互联网灾后寻人需求应运而生。

灾后寻人项目前期具有的功能是灾区中的2类用户上报信息：

一类是健康平安的用户上报自己的信息；

一类是自己无法上报，需要他人帮助其上报信息。

寻人的用户可以通过发布寻人信息以及搜索寻人2种方式在收集来的信息数据中匹配自己需要寻找的人。

灾后寻人项目后期拓展可能会链接其它互联网寻人平台的数据信息，这样更有利于数据信息的共享。这样才算真正意义上秉承了互联网开放、共享、互联的理念。

# 实现的功能

1. 发生灾难后，用户希望寻找灾难地区的某个人；
2. 发生灾难后，灾区的用户能自主发布平安报告
3. 发生灾难后，救助人可以在平台上发布被救助者的相关信息，供他人查询

# 设计考虑

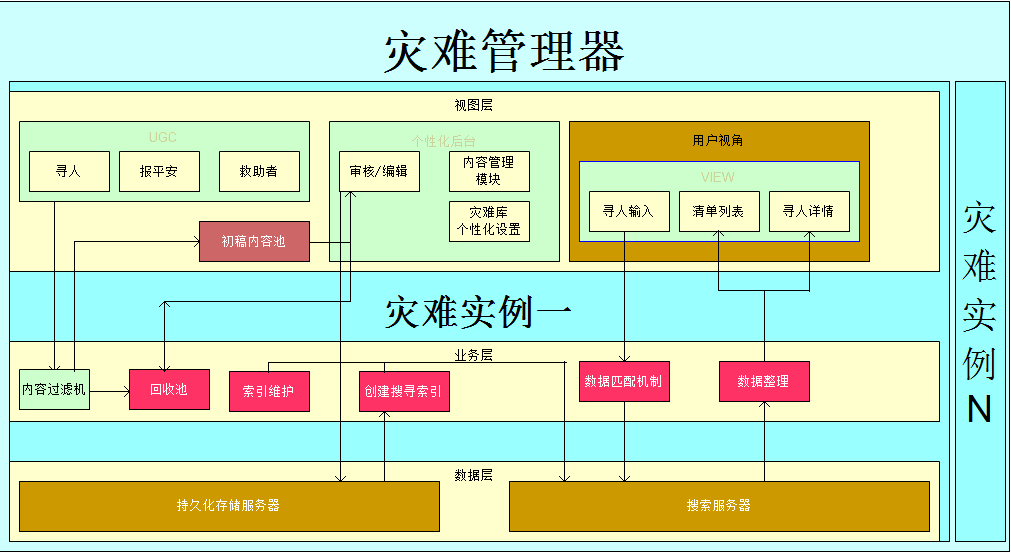
3.1 设计模式

灾后寻人将用于各种场景下的寻人发布与查询，在统一数据结构的情况下，需要能适应各种情景下的使用，所以会采用MVC模式分离视图、逻辑和数据层，考虑项目开发时间限制，一期会以实现业务需求为主、拓展冗余为辅。

3.2 运行环境

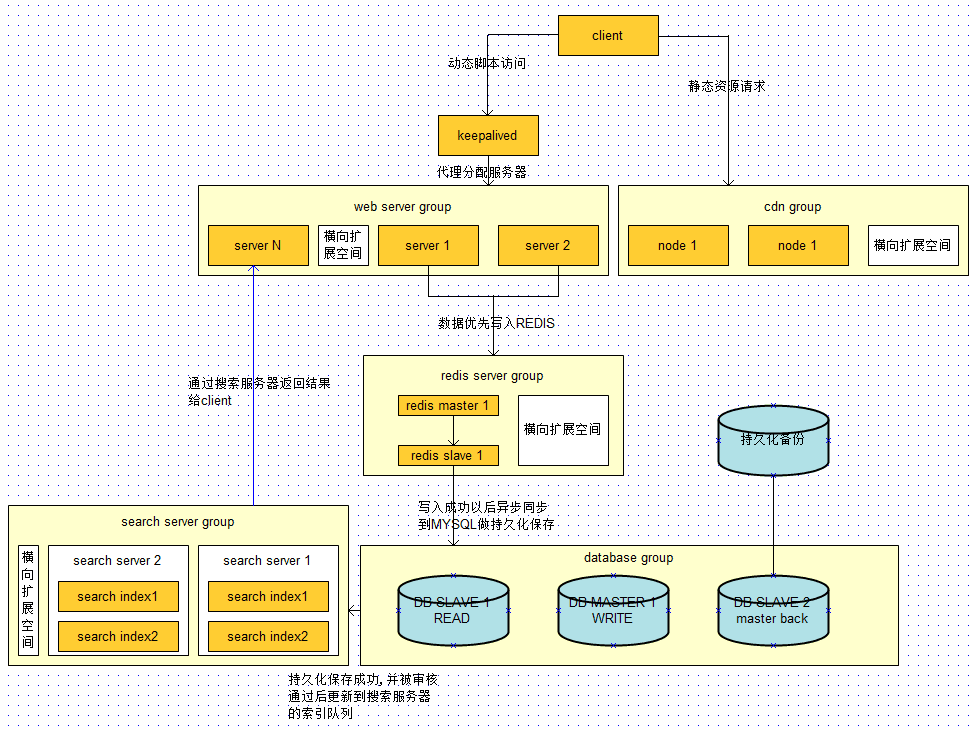
centos 6.4+, nginx 1.2+, php 5.4+, mysql 5.1+, redis3.x, xunsearch1.4.X

3.3业务流程/结构图



3.4系统架构图

可热机横向扩展及冗余备份，以保证业务系统在突发事件高并发的情况下正常运作，事件冷淡期退出冗余服务器以节省资源。



### 3.5 语言选择

后端使用PHP、前端使用HTML5+JS

### 3.6 WEB框架

尽量保证轻量级不使用框架，前端采用JQUERY标准库

### 3.7 API设计规范

基于HTTP协议的API接口

### 3.8 协议

HTTP1.0

### 3.9 接口原则

后端程序尽量组合成聚合数据提供给接口，最大限度简化接口数量，以便后期维护和拓展。

### 3.10 安全性

存储定期冷备份，数据传输通过密钥加密

## 4 详细设计

4.1数据表设计（仅列出主要数据表）

|  |
| --- |
| 1. base\_features(基本特征表)   `id` INT( 6 ) **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY** **COMMENT** '索引',  `type` TINYINT( 2 ) **NOT** **NULL** **COMMENT** '特征类型（在config.inc.php中配置，特征类型暂不提供管理功能'）,  `tag` VARCHAR( 20 ) **NOT** **NULL** **COMMENT** '特征标签名',  `defined` TINYINT( 1 ) **NOT** **NULL** DEFAULT '0' **COMMENT** '是否默认加载类型0=否 1=是',   1. `disa\_base`（灾难事件表）   `id` INT( 10 ) **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY** **COMMENT** '索引',  `disname` VARCHAR( 50 ) **NOT** **NULL** **COMMENT** '事件名称',  `status` TINYINT( 1 ) **NOT** **NULL** DEFAULT '0' **COMMENT** '状态 0 = 未进行 1=进行中',  `timeline` INT( 10 ) **NOT** **NULL** **COMMENT** '创建时间',  `startime` INT( 10 ) **NOT** **NULL** **COMMENT** '开始进行时',  `endtime` INT( 10 ) **NOT** **NULL** **COMMENT** '结束时',  `pnum` INT( 10 ) **NOT** **NULL** DEFAULT '0' **COMMENT** '总人数',  `dnum` INT( 10 ) **NOT** **NULL** DEFAULT '0' **COMMENT** '成功人数',  `image` VARCHAR( 200 ) **NOT** **NULL** **COMMENT** '封面背景图'   1. disa\_features(灾难事件特征表，该表数据在创建灾难事件时从base\_features中导入)   `id` INT( 10 ) **NOT** **NULL** **PRIMARY** **KEY** **COMMENT** '特征ID',  `did` INT( 10 ) **NOT** **NULL** **COMMENT** '灾难事件ID',  `type` TINYINT( 2 ) **NOT** **NULL** **COMMENT** '特征类型‘,  `tag` VARCHAR( 20 ) **NOT** **NULL** **COMMENT** '特征文字描述',  `status` TINYINT( 1 ) **NOT** **NULL** DEFAULT '1' **COMMENT** '是否启用-该字段主要预防事件进行中需要对特征进行删除时，涉及数据关联的情况下做置0处理',   1. user\_baseinfo   `uid` INT( 10 ) **NOT** **NULL AUTO\_INCREMENT** **AUTO\_INCREMENT** ,  `did` INT( 10 ) **NOT** **NULL** **COMMENT** '灾难事件ID',  `name` VARCHAR( 20 ) **NOT** **NULL** **COMMENT** '姓名 index',  `sex` TINYINT( 1 ) **NOT** **NULL** **COMMENT** '性别',  `age` TINYINT( 2 ) **NOT** **NULL** **COMMENT** '年龄',  `timeline` INT( 10 ) **NOT** **NULL** **COMMENT** '入库时间',   1. user\_extend (被寻者扩展数据表)   `uid` INT( 10 ) **NOT** **NULL** **COMMENT** '对应base uid',  `mobile` VARCHAR( 15 ) **NOT** **NULL** **COMMENT** '联系电话 index',  `trait` VARCHAR( 20 ) **NOT** **NULL** **COMMENT** '个人特征的ID序列持久化存储',  `photo` VARCHAR( 50 ) **NOT** **NULL** **COMMENT** '个人照片的ID序列持久化存储',  `status` TINYINT( 1 ) **NOT** **NULL** **COMMENT** '发布的状态0=未审核 1=正常 2=已有匹配',  `address` VARCHAR ( 10 ) **NOT** **NULL** **COMMENT** '目前所在地',  `remark` VARCHAR( 255 ) **NOT** **NULL** **COMMENT** '备注说明',   1. user\_helperinfo(帮助者数据表)   `id` INT( 10 ) **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY** ,  `name` VARCHAR( 30 ) **NOT** **NULL** **COMMENT** '救助者姓名',  `mobile` VARCHAR( 20 ) **NOT** **NULL** **COMMENT** '联系电话',  `idcode` VARCHAR( 18 ) **NOT** **NULL** **COMMENT** '身份证号',  `did` INT( 10 ) **NOT** **NULL** **COMMENT** '被救助人的ID',   1. admin\_userinfo(平台编辑用户表)   `id` INT( 10 ) **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY** ,  `username` VARCHAR( 30 ) **NOT** **NULL** **COMMENT** '帐号名',  `password` VARCHAR( 32) **NOT** **NULL** **COMMENT** '密码',  `halt` CHAR( 5 ) **NOT** **NULL** **COMMENT** '密码加密盐',  `nicename` VARCHAR( 18 ) **NOT** **NULL** **COMMENT** '对外显示的昵称',  `lastlogin` INT( 10 ) **NOT** **NULL** **COMMENT** '最后登录的时间',  `supers` TINYINT( 1 ) **NOT** **NULL** **COMMENT** '是否具有全局权限',  `loginum` INT( 5 ) **NOT** **NULL** **COMMENT** '登录的次数',   1. admin\_userlog(平台编辑用户日志表)   `id` INT( 10 ) **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY** ,  `uid` VARCHAR( 30 ) **NOT** **NULL** **COMMENT** '编辑ID',  `timeline` INT( 10) **NOT** **NULL** **COMMENT** '发生的时间',  `action` VARCHAR( 15 ) **NOT** **NULL** **COMMENT** '控制器名',  `model` VARCHAR( 15 ) **NOT** **NULL** **COMMENT** '操作名',  `cid` INT( 10 ) **NOT** **NULL** **COMMENT** '对应的灾难ID',  `status` TINYINT( 1 ) **NOT** **NULL** **COMMENT** '是否成功',   1. admin\_channel(平台编辑用户日志表)   `id` INT( 10 ) **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY** ,  `uid` VARCHAR( 30 ) **NOT** **NULL** **COMMENT** '编辑ID',  `cid` INT( 10) **NOT** **NULL** **COMMENT** '灾难ID',  `status` TINYINT( 1 ) **NOT** **NULL** **COMMENT** '是否被锁定', |

#### 5前端部份整体设计

5.1 用户部份UI

包含3个主要JS文件

jquery.js JQUERY基础库

main.min.js 主要的业务方法类库

base.min.js 基于JQUERY自定义开发的扩展类库

包含四个主要页面，页面会有多个层级进行切换以保证操作中的流畅性。单个页面中的各类数据会通过AJAX接口从服务端获取。

Index.html 聚合页

main.html 表单页

List.html 列表页

View.html 详情页

5.2 用户的操作流程

每个灾难事件的用户是通过不同入口进来，进来以后只能看见当前入口所指定的灾难事件。

可通过聚合页的入口进行信息发布，发布成功以后用户可以立即查看到自己发布的内容和所匹配的数据（如果有的话）

用户的数据要在后台经过编辑审核过后才能被其他人搜索匹配出来，列表页同理。

5.3 前端主要接口及说明

/index

聚合页接口，获取当前事件所包含的相应数据（具体见前端需求的首页内容）

/index/uploader 用户上传图片接口，一期仅支持单实例上传

提交方式：POST

提交数据：image data

返回结果：{status:1, url:’’, timeline:’’}

/index/main 提交一个新的寻人信息数据（包含报平安,救助者)

提交方式：POST

提交数据：add user form, data\_type

返回结果：{status:1, result:{uid:{}, uid:{}}} 其中result的数据为当前系统内已经有的匹配内容，如果超过5个就只返回前5个

/index/list 获取已经存在寻人信息列表

提交方式：GET

提交数据: page\_num, search\_key

返回结果:{status:1, data:{id:{}, id:{}}}

/index/view 获取某个寻人信息的详情

提交方式:GET

提交数据：user\_id

返回结果：{status:1, userinfo:{}, result:{}} 其中result表示比较接近的三条匹配结果