|  |
| --- |
| NJU CS |
| 赈灾管理系统 概要设计说明书 |
| TEAM 18 |
| Version 1.0 |
| **121220312徐栋 121220313许迪文 121220309李乾科 121220052李昀 121220070缪静文** |
|  |
| **2015/3/17** |

目录

[一、引言 3](#_Toc417656355)

[1.1 介绍 3](#_Toc417656356)

[1.2 版本信息 3](#_Toc417656357)

[1.3 参考资料 3](#_Toc417656358)

[二、框架设计 3](#_Toc417656359)

[2.1 需求/功能规定 3](#_Toc417656360)

[2.2 项目平台框架、运行环境 4](#_Toc417656361)

[2.3 平台的运行模式 4](#_Toc417656362)

[2.4 设计理念 5](#_Toc417656363)

[2.5 功能与模块关系 5](#_Toc417656364)

[三、数据库设计 6](#_Toc417656365)

[3.1 命名规则 6](#_Toc417656366)

[3.2 数据库各表单设计 6](#_Toc417656367)

[四、WEB平台设计 7](#_Toc417656368)

[4.1 说明 7](#_Toc417656369)

[4.2 WEB平台页面设计 7](#_Toc417656370)

[五、APP设计 9](#_Toc417656371)

[5.1 说明 9](#_Toc417656372)

[5.2 通信接口 9](#_Toc417656373)

[5.3 界面设计 9](#_Toc417656374)

[六、接口设计 9](#_Toc417656375)

[6.1 用户接口 9](#_Toc417656376)

[6.2 外部接口 9](#_Toc417656377)

[6.3 内部接口 9](#_Toc417656378)

[七、系统故障应对 10](#_Toc417656379)

[7.1 错误处理表 10](#_Toc417656380)

[7.2 系统维护 10](#_Toc417656381)

[八、附录 11](#_Toc417656382)

[8.1 WEB页面设计草稿 11](#_Toc417656383)

[8.2 APP页面设计草稿 12](#_Toc417656384)

# 一、引言

## 1.1 介绍

本文档主要针对项目的概要设计说明书，详细的设计项目的框架、接口、实现方式等，对项目的各个组成部分提出了详尽的设计方案。

本文档的作用是为编码实现提供框架、为修改拓展提供依据、为项目控制审核提供参考。预期的读者为代码编写者、项目管理人员、项目拓展维护工程人员。

本项目为开发一个赈灾管理系统，主要用于赈灾管理中的资源调度、志愿者调度、灾民收容与管理。同时本系统可以在较小开发成本下，转换为任意资源调度、人员管理的系统。项目名为Amalthea。当下的赈灾管理多采用人工管理的原始方式。没有系统完善的管理软件，管理需要用人工操作数据库，建立和修改费时费力，难以达到令人满意的速度，对二次灾害的发生应对也不够迅速，并且由于数据库临时建立，稳定性堪忧，误操作可能性也较大。同时灾区的通信受到严重破坏，即使付出极高的代价，抢修完成后，覆盖范围还是不广，性能不够强大，所能提供的QoS有限，同时抢修的费用也十分高昂。

本项目为南京大学软件工程课程项目，由徐栋、李乾科、许迪文、李昀、缪静文协同完成。

## 1.2 版本信息

Version 1.0（2015/4/23）：

为设计文档的第一版，详细设计了项目的框架、接口、实现方式。

## 1.3 参考资料

概要设计说明书（GB8567——88）模板。

# 二、框架设计

## 2.1 需求/功能规定

系统的需求（即所要实现的功能）规定如下：

2.1.1添加灾民及信息、状态

2.1.2修改灾民信息、状态

2.1.3查询灾民信息

2.1.4用户注册

2.1.5志愿者申请、信息录入

2.1.6志愿者审核

2.1.7志愿者当前状态修改

2.1.8任务请求/援助请求

2.1.9任务创建、信息录入

2.1.10任务分配（分配执行者/志愿者）

2.1.11任务确定/终止

2.1.12任务当前状态修改

2.1.13庇护所创建、信息录入

2.1.14庇护所信息修改

2.1.15庇护所信息查询

2.1.16物资捐献

2.1.17物资捐献确认

2.1.18物资状态修改（通过任务创建、分配修改）

2.1.19灾区创建、信息录入

2.1.20灾区信息查询

2.1.21灾区信息发布、修改

2.1.22损失统计

2.1.23登陆/注册

2.1.24 APP下载

2.1.25 APP额外功能

## 2.2 项目平台框架、运行环境

平台总共由四个大的模块组成：WEB服务器（各页面为子模块）、数据库、服务器支撑程序、APP终端。

🡪WEB服务器实现的功能包括：2.1.1至2.1.6，2.1.8至2.1.22。

🡪数据库主要提供数据的存储、管理。

🡪服务器支撑程序主要与APP终端通信、处理与APP相关的数据。

🡪APP终端的功能包括：2.1.3至2.1.5，2.1.7，2.1.8，2.1.12，2.1.15 ，2.1.20 ，2.1.23，2.1.25。

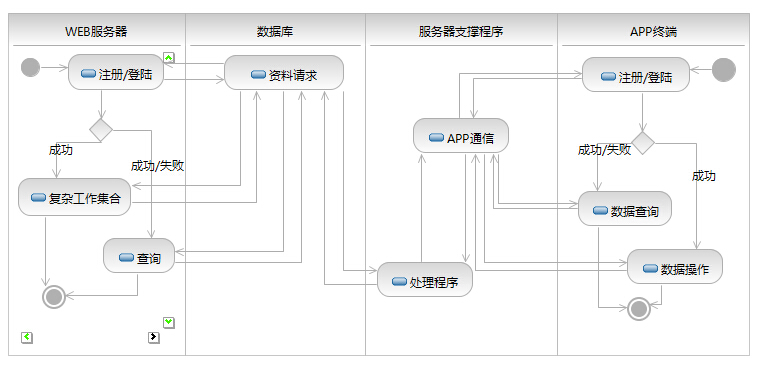
WEB平台的运行环境

🡪软件：WIN7+XAMPP或Ubuntu+XAMPP，数据库为MYSQL

🡪硬件：CPU intel i5 2430 2.4GHz\*2、内存DDR3 4G、硬盘5400转

## 2.3 平台的运行模式

平台的一个简单的运行模型如图2.3.1所示，数据库提供数据层面的支持，WEB页面可以直接操纵数据库，但APP与数据库的数据交互需要经由一个服务器支撑程序进行。同时服务器支撑程序还会进行一些额外的数据处理，比如消耗统计等等。

图2.3.1平台运行模型

WEB服务器的两个主要管理界面，areadomain（灾区管理）、shelterdomain（庇护所管理）的运行模型如图2.3.2和图2.3.3。这两个页面实现了平台核心的数据操纵功能，包括人口管理、志愿者管理、援救请求及处理、救援任务、庇护所、物资库存、消息发布等。

灾区管理页面与庇护所管理页面的操作逻辑相同，通过控制选项选择管理组件，然后在相应的管理组件内进行管理操作，相应的操作会产生一系列的数据操纵指令，最后通过PHP的数据操纵模块反映到数据库中。



图2.3.2 灾区管理运行模型

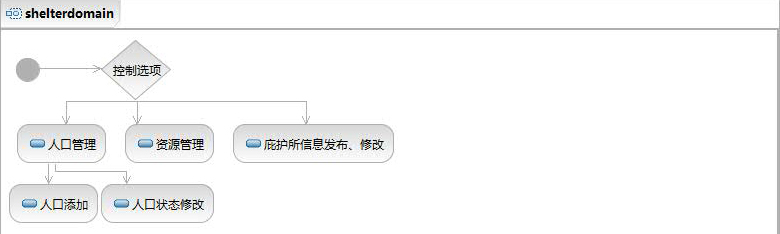


图2.3.3 庇护所管理运行模型

## 2.4 设计理念

在系统平台的选择上，我们选择了WEB平台，WEB平台相对于JAVA平台来说具有几个优点：界面美观、事件处理方便、布局自由度高、跨平台、兼容性强等。同时针对于这个系统的需求来说，WEB平台更适合做灾区信息的传播，同时更方便用户使用，免去了客户端安装的麻烦。

同时我们增加了APP终端，这是因为志愿者在完成任务的时候，使用APP完成一些基本的操作会更加快捷且方便，这在实际应用中是十分具有意义的，可以极大的提高志愿者的工作效率。

## 2.5 功能与模块关系



# 三、数据库设计

## 3.1 命名规则

采用26个英文字母和0－9这十个自然数，加上下划线\_组成，共37个字符。不出现其他字符。鉴于大小写规则难记忆，而且有些情况下，数据库不区分大小写，统一采用大写字母。

采用英文单词或英文短语（包括缩写）作为名称，不使用无意义的字符或汉语拼音。

 名称应该清晰明了，能够准确表达事物的含义，最好可读，遵循“见名知意”的原则。

## 3.2 数据库各表单设计

**3.2.1 CITIZEN（公民）**

描述：所有与本次灾害相关的公民信息。

字段：

1. ID：身份证号，STRING类型，18位，**主键**；
2. NAME：姓名，STRING类型；
3. SEX：性别，STRING类型，双选：男、女；
4. RACE：民族，STRING类型；
5. HOME：籍贯，STRING类型；
6. BLOODTYPE：血型，STRING类型；
7. LEVEL：权限等级，INT型，（0为默认级别，普通用户，不记录在数据库中），1为灾民，2为志愿者，3为管理员，4为高级管理员，5为庇护所管理员，6为最高管理员。

**3.2.2 VICTIM（灾民）**

描述：所有受到本次灾害影响的当地居民。

字段：

1. VNO：灾民序号，INT型，从1开始排列，**主键**；
2. ID：身份证号，**主键**；见上方
3. VSTATUS：状态，STRING类型，四选：死亡、受伤、失踪、其他；
4. VFAMILY：亲属及联系方式，STRING型。

**3.2.2.1 MISSING（失踪灾民 ）**

描述：所有失踪灾民。

字段：

1. VNO：灾民序号，**主键**；见上方
2. MTIME：失踪时间，DATE型，日期；
3. MPLACE：失踪地点，STRING型；
4. MREMARK：备注，STRING型，包括但不限于失踪时的衣着。

**3.2.2.2 INJURED （受伤灾民）**

描述：所有受伤灾民。

1. VNO：灾民序号，**主键**；见上方
2. INJURY：受伤程度，STRING型，三选：轻伤、重伤、痊愈；
3. IPLACE：就医医院（或转移中），STRING型；
4. IREMARK：备注，STRING型，包括但不限于伤情描述。

**3.2.2.3 DEAD （死亡灾民）**

描述：所有死亡灾民；

1. VNO：灾民序号，**主键**；见上方
2. DTIME：死亡时间，STRING型，严格来说是尸体发现时间；
3. DPLACE：死亡地点，STRING型，严格来说是尸体发现地点；
4. CORPSE：尸体描述，STRING型，包括但不限于尸体停放位置；
5. DREMARK：备注，STRING型，包括但不限于尸检报告（确切死亡时间）。

**3.2.3 VOLUNTEER（志愿者）**

描述：所有参加救灾活动的人员，包括灾区管理人员。

字段：

1. VONO：志愿者序号，INT型，从1开始排列，**主键**；
2. ID：身份证号，**主键**；见上方
3. VOSTATUS：状态，STRING型，四选：申请中、未就绪、就绪、工作；
4. VOHEIGHT：身高，INT型，以CM为单位；
5. VOWEIGHT：体重，INT型，以KG为单位；
6. VOEDU：教育状况，STRING型，包括但不限于学历及专业；
7. VOPLACE：当前地点，STRING型；
8. VOFAMILY：亲属及联系方式，STRING型。

**3.2.4 ADMINISTRATOR（管理员）**

描述：系统管理员。

字段：

1. ADNO：管理员编号，INT型，从1开始排序，**主键**；
2. VONO：志愿者编号，**主键**；见上方
3. BOSSNO：上级管理员编号，INT型，没有则为0；
4. ADREMARK：备注，STRING型，包括但不限于职位。

**3.2.5 Task（任务）**

描述：救灾工作所有任务信息。

字段：

1. TNO：任务编号，INT型，从1开始排序，**主键**；
2. TNAME：任务名称，STRING型
3. ADNO：管理员编号；见上方
4. TSTATUS：任务状态，STRING型，三选：未开始、进行中、已结束。

**3.2.3 Shelter（庇护所）**

描述：

字段：

ShNO：

**3.2.4 Supplies（物资）**

描述：

字段：

SuNO：

**3.2.5 Area（灾区）**

描述：

字段：

ANO：

# 四、WEB平台设计

## 4.1 说明

WEB平台为本系统的主控组件，大多数的数据操纵是在WEB平台上完成的，WEB平台实现了在2.2.1节中为WEB平台分配的功能。下面描述WEB平台的各组成模块（即各个页面）的设计，其实现功能与页面关系的直观二维表在2.4节中阐述。在附录中，我们给出了几个主要界面的设计草图。

用户组及用户权限：

🡪未注册用户：不可访问areaadd、areadomain、shelterdomain、donate、user

🡪注册普通用户：不可访问areaadd、areadomain、shelterdomain

🡪志愿者：不可访问areaadd、areadomain、shelterdomain

🡪灾区管理员：不可访问areaadd

🡪庇护所管理员：不可访问areaadd、areadomain

🡪高级管理员：都可以访问

## 4.2 WEB平台页面设计

**4.2.1 index主页**

Index是WEB平台的主页面，主要提供整个网站的简要介绍、与网站相关的新闻消息以及相关的链接。

其中平台简介为页面正中的2/3大小区域，其内容为切合网站主题（即赈灾管理系统）的图片及文字描述，同时提供重要的灾区新闻。功能链接入口及APP下载位于页面下方1/3的区域。

**4.2.2 header页头**

Header是位于所有界面顶端的部分，主要提供页面导航作用以及登陆信息的显示。

Header总共只有一行，行的高度约为50-60px。从左到右分别有 LOGO按钮，平台页面导航菜单（主页、灾区列表、查询、捐献、About等），登陆/注册按钮，用户信息。其中LOGO为一个图片，可以通过点击链接到主页；用户信息可以链接到用户后台界面。



**4.2.3 footer页脚**

Footer是位于所有界面底部的部分，主要展示页面信息等。

**4.2.4 arealist灾区列表**

Arealist是灾区列表页面。页面分栅栏，为N\*M的结构，其中N=3，可以进行微调。其中每一个“格子”为一个灾区的简要介绍，每个“格子”中，上方是图片并在底部显示灾区名、灾区地址等简要内容，点击后可以进入相应的灾区页面（area）。

若权限达到（即系统的高级管理员），在最后一个“格子”后额外显示一个添加（新建）灾区的格子，样式设计为一个加号黑白图片，可以点击连接到areaadd进行灾区的添加。

**4.2.5 areaadd灾区添加**

Areaadd是灾区添加页面，在页面居中位置，显示一张详细的表单，下方为一个提交按钮。通过表单填写灾区的基本信息，在数据库中插入数据。

**4.2.6 area灾区信息**

Area页面为平台主要的界面之一，每一个灾区均有一个area页面，由php动态生成。Area页面包括灾区简要介绍，顶端1/3区域有一个滚动的图片栏、为灾区实景。下面2/3区域为与此灾区相关的详细信息，为多个表单的形式：包括资源、人口状态（包括失踪人口名单统计等）、庇护所状态（例如XX庇护所，地址，安顿人口数量，物资充足状态（充足，正常，紧缺））。

因为页面的信息较多，对于某类信息比如说某条新闻展开的话，不考虑新开一个页面，而是使用JS动态的更改页面的信息展示。

**4.2.7 areadomain灾区管理后台**

Areadomain页面与area页面成对出现，每个灾区的管理员都可以进入相应的灾区管理后台。界面的组织为左1/3为管理事项列表，点击相应的管理事项，在右2/3会显示相应的管理器组件。管理事项在图2.3.2中形象描述，包括任务请求队列，任务进度，新建任务，添加庇护所，审核志愿者，物资调度，新闻发布等。

**4.2.8donate捐献物资**

Donate为捐献页面，捐赠人可以填写捐赠物品的信息及运单号，在登陆后，可以在此页面查看每一笔捐赠的状态。

**4.2.9 user用户后台**

初步设计中只提供志愿者申请操作、管理人员的相应信息显示及相应管理界面的链接。

**4.2.10 shelter庇护所信息查询**

Shelter为庇护所信息显示界面。设计方案有二：其一是设计为一个单独的页面，用于显示某庇护所的详细信息，包括实景图片、容纳人员、资源状况、设施配置等；其二是设计在灾区信息area页面，使用JS动态的显示这些信息。具体的方案选择要根据需要展示的信息的数量决定。

**4.2.11 shelterdomain庇护所管理**

为庇护所提供庇护所的管理操作，与shelter成对出现，其布局格式与areadomain类似，左边为控制列表，右边为管理操作面板。管理事项在图2.3.3中形象描述，包括人员登记管理、庇护所当前资源管理、庇护所信息的发布与修改。

**4.2.12 login登陆/注册**

在未登陆情况下，在header中才会显示登陆/注册的连接，login界面默认显示登陆表单，下方有按钮选择“没有账号，注册”，点击后显示注册表单。表单的位置为左1/2区域，右1/2显示平台新闻及注册提示等信息。

**4.2.13 about系统介绍**

对系统详细的介绍。

**4.2.14 request任务请求**

任务请求是一个表单，通过填写表单来申请一个援助请求，这个请求会在areadomain（灾区管理后台）中看到并有灾区管理员审核。

# 五、APP设计

## 5.1 说明

开发系统：Android 4.3

用户组及用户权限：

🡪普通用户：无需注册，安装app即可使用

🡪志愿者：需要申请，使用app时需要登录

功能服务：

2.1.3 查询灾区人员

2.1.4 用户注册

2.1.5 志愿者申请

2.1.7个人状态修改

2.1.8 任务请求

2.1.12 任务状态修改

2.1.15 庇护所信息查询

2.1.20 灾区信息查询

2.1.25 定位

## 5.2 通信接口

使用流式sokcet与服务器支撑程序通信，封装相关的socket通信函数。

## 5.3 界面设计

打开app -->普通界面

普通界面 -->志愿者界面（已注册登陆）or 登陆界面（未登录）

普通界面 -->庇护所信息界面、查询人员界面、地图显示界面

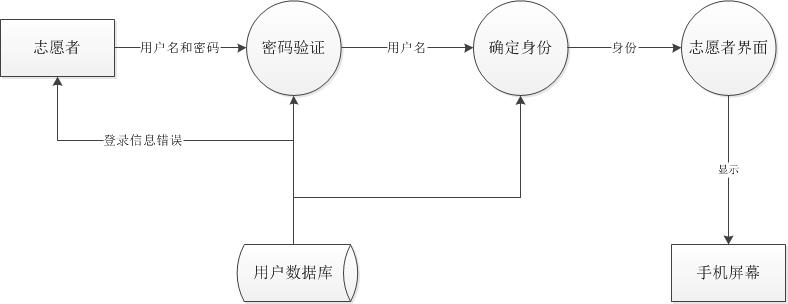
志愿者界面 -->任务界面

## 5.4 APP数据流

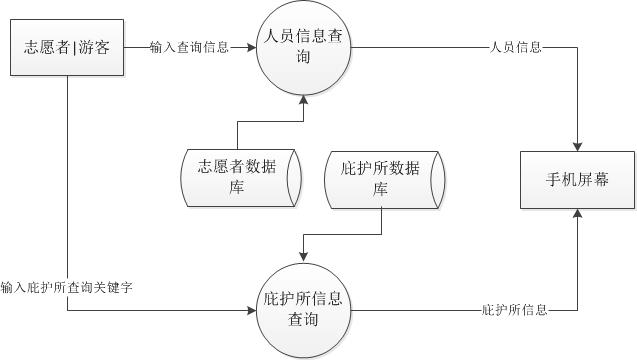


## 5.5 APP活动图

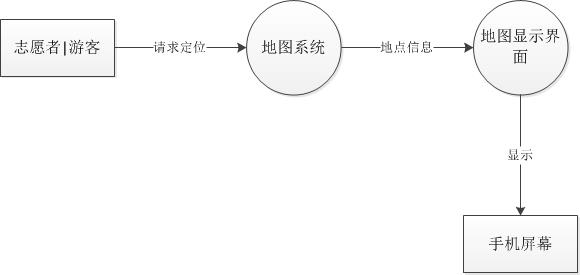
**5.5.1 登录子系统**



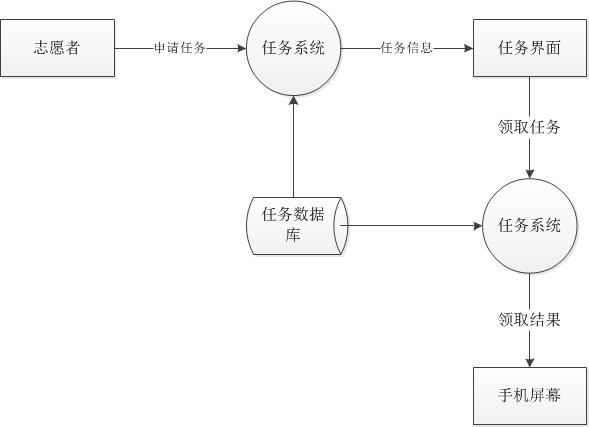
**5.5.2查询子系统**



**5.5.3定位子系统**



**5.5.4任务子系统**



# 六、接口设计

## 6.1 用户接口

本平台的两部分WEB平台与APP终端中提供与用户的交互，这两部分都是可视化的，用户可以通过手机屏幕、手机键盘、计算机键盘、计算机鼠标、计算机显示器与系统进行交互。

## 6.2 外部接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 鼠标 | 键盘 | 网络 | 手机 | 服务器设备 |
| WEB平台 | Yes | Yes | Yes | Yes/No | Yes |
| APP终端 | No | No | Yes | Yes | No |
| 服务器支撑程序 | No | No | Yes | No | Yes |
| 数据库 | No | No | Yes | No | Yes |

## 6.3 内部接口

在下表6.3.1中，第一列中的模块为发送数据方，第一行中的模块为接受数据方。表6.3.2中，是在非导航栏中的页面跳转或子模块调用。第一列中为链接起始页面或父模块，第一行中为链接目的页面或子模块。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | APP | 服务器支撑程序 | 数据库 | WEB平台 |
| APP | —— | 请求数据/修改数据 | —— | —— |
| 服务器支撑程序 | 数据/修改结果 | —— | 数据库操作 | —— |
| 数据库 | —— | 数据/修改结果 | —— | 数据/修改结果 |
| WEB平台 | —— | —— | 请求数据/修改数据 | —— |

表6.3.1 模块之间接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Area | Shelter | Query | Areadomain | Shelterdomain |
| Arealist | \* |  |  |  |  |
| Area |  |  | \* |  |  |
| Shelterlist |  | \* |  |  |  |
| User |  |  |  | \* | \* |
| Shelter |  |  | \* |  |  |

表6.3.2 子模块跳转调用接口

# 七、系统故障应对

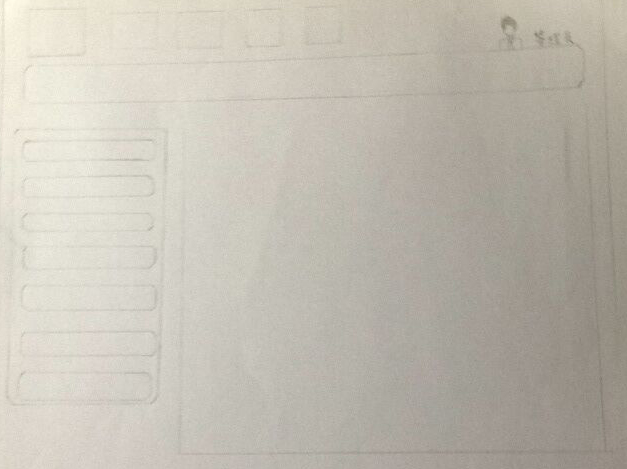
## 7.1 错误处理表

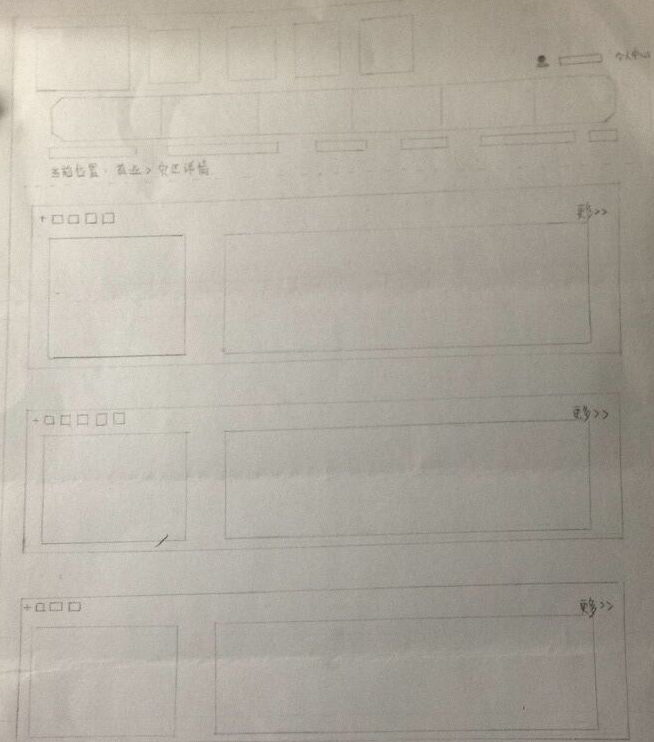
|  |  |  |
| --- | --- | --- |
| 错误信息 | 错误原因 | 应对方案 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 7.2 系统维护

# 八、附录

## 8.1 WEB页面设计草稿





## 8.2 APP页面设计草稿

