



校园帮失物招领平台 设计说明书

系统设计说明书

目录

1 引言.....	4
1.1 编写目的.....	4
1.2 名词术语.....	4
1.3 参考资料.....	4
2 总体设计.....	4
2.1 建设背景.....	4
2.2 系统建设目标.....	5
2.2.1 提高用户体验.....	5
2.2.2 建立统一的 MIS 应用架构.....	5
2.2.3 后台管理平台.....	6
2.2.4 用户信息交流平台.....	7
2.3 设计和实现约束.....	7
2.4 组织结构和用户类.....	7
2.5 系统架构.....	9
2.5.1 技术架构.....	10
2.5.2 功能架构.....	11
2.5.3 UI 架构.....	12
2.6 接口设计.....	14
2.7 系统环境.....	15
2.7.1 硬件环境.....	15
2.7.2 软件环境.....	15
2.8 非功能特性设计.....	16
2.8.1 系统兼容性.....	16
2.8.2 安全性.....	16
2.8.3 运行效率.....	16
2.8.4 可扩展能力.....	16
3 系统公共模块设计.....	16
3.1 权限控制.....	16
3.2 日志处理模块.....	17
3.2.1 日志记录规范.....	17
3.2.1.1 日志分类.....	17
3.2.1.1.1 系统业务运行日志.....	17
3.2.1.1.2 系统调试日志.....	18
3.2.1.2 日志级别分类.....	18
3.2.1.3 规范.....	19
3.3 异常处理模块.....	21
3.3.1 规范.....	21
3.3.2 建议.....	22
3.3.3 异常处理模块.....	22
3.4 Session 处理.....	23

系统设计说明书

3.5 UI 框架设计例子.....	24
微信小程序设计 ui 框架:.....	24
4 应用基础框架子系统设计.....	26
4.1 系统框架.....	26
4.1.2 概述.....	26
4.1.2 用例描述.....	27
4.1.2.1 权限管理的用例描述.....	28
4.1.3 组织机构管理的用例描述.....	31
角色管理.....	31
4.1.4 实体关系描述（E-R 图）.....	36
4.2 构件包设计.....	36
4.2.1 构件包列表.....	36
4.2.2 构件包关系图.....	37
4.2.3 基础服务构件包.....	37
4.2.3.1 概述.....	37
4.2.3.2 流程设计.....	37
4.2.3.2.1 信息发布流程.....	37
4.2.3.2.1 信息浏览流程.....	38
4.2.3.3 功能列表.....	39
4.2.3.4 功能设计.....	39
4.2.3.4.1 信息查找.....	39
4.2.3.5 构件包的服务接口.....	41
4.2.3.5 构件包的引用接口.....	42

系统设计说明书

1 引言

1.1 编写目的

本设计说明书文档包括该项目的建设背景、目标、建设内容、系统架构、接口、数据模型、功能模型、部署模型、功能设计等的描述，用于指导该项目的开发与部署，同时，作为该项目的重要技术资料，作为系统未来维护或扩展的参考。

本文档的读者为本系统的设计、开发人员、接口系统的开发人员、系统维护人员。

1.2 名词术语

mini apps	小程序，mini Application 的缩写，一种不需要下载安装即可使用的应用。
后台管理平台	
系统管理环境	
MIS	管理信息系统（Management Information System）
后台管理平台	

1.3 参考资料

《系统设计说明书范文.doc》

2 总体设计

2.1 建设背景

路不拾遗尽然可贵，但在现实社会中，我们更加提倡的是拾金不昧。我们每个人都会丢失东西，无论我们丢失什么，我们都有一个同样的希望：丢失的东

系统设计说明书

西可以回到自己手中！

在学校生活中，谁都难免会有马虎的时候，会有一不小心丢失东西的时候，这个时候就失物招领的作用就体现了出来，一个在学校内部的失物招领，可以最大限度的找回丢失的物品。但是，在学校中丢失的物品或捡到的物品只有到表白墙发表或者红楼那边挂起，这两种方法都不能最大限度的将失物招领的信息发表出去，也不能实时的注意到自己丢失的物品是否有人找到，甚至还会出现冒领的情况。所以，如果在学校中有一个专门进行失物招领的平台，一个可以方便发表消息和查看信息的平台，我们的校园生活是不是会更加轻松一些。通过开发设计小程序解决在高校中出现丢失东西时给人们带来的困扰，提高找回效率。

2.2 系统建设目标

2.2.1 提高用户体验

希望提供给用户：

- UI 展现、操作友好
 - 易交互：交互性强，放弃占存储空间大，繁琐的 app 而是使用占存储空间小的 mini apps；
 - 丰富的控件：开发或集成丰富的 UI 控件，既丰富了用户的交互手段，又方便了开发人员。
- 程序内容
 - 一次验证：用户登录后进行一次学号认证；
 - 一个消息平台：提供给用户一个可以对丢失物品和捡到物品发表信息的平台，同时可以进行信息的浏览和查找。

正是基于上述原因，要求本系统实现：

- 支持 RIA 的 UI 框架；
- 微信小程序 MIS 界面。

2.2.2 建立统一的 MIS 应用架构

希望给程序提供：

- 用户模型及服务
 - 用户模型：收集学生信息，整合验证信息，建立用户模型；

系统设计说明书

- 权限模型：通过学生证验证完成学生到用户的权限升级。
- 集中流程信息处理
 - 发布信息：抓取用户在小程序中发布的信息，统一展现给用户处理；
 - 信息浏览：抓取用户在小程序中的已经发布过的信息，供用户查看；
 - 审核结果：抓取管理员的审核结果，供用户查看；
 - 用户发布信息：通过用户学号在已经发布过的信息中抓取信息，供用户查看。

正是基于上述原因，要求本系统实现：

- 应用基础框架；
- 集中信息交流中心；
- 用户管理中心。

2.2.3 后台管理平台

以上目标有机形成后台管理平台，提供校园帮后台管理的统一入口：



图 1

系统设计说明书

2.2.4 用户信息交流平台



图 2

2.3 设计和实现约束

通过项目启动前的方案验证和技术预研工作，为本项目的实施打下了良好的基础，并确定了如下的设计和实现原则：

- 技术架构采用 J2EE+普元 E0S6
- UI 框架采用微信小程序设计规范
- 小程序界面采用 Portal 实现
- 用户认证采用 LDAP

进行本小程序的实现约束条件如下：

- 开发周期短：三个月的开发时间需要开发者合理规划时间，做到多项任务并发。
- 所采用的方法与技术有限：项目团队成员的技术水平不够成熟，需要在开发中并发学习多种技术和能力。
- 团队是第一次面临次任务，所以团队之间磨合并没有那么好。

2.4 组织结构和用户类

(1) 后台管理员：登录后台管理网站，通过后台对发帖信息、审核信息、回应信息和用户人员等方面的管理。

用户：登录微信小程序，通过平台对信息进行浏览，查看他人发布的帖子信息或自己进行信息发布。

系统设计说明书

(2) 图表如下

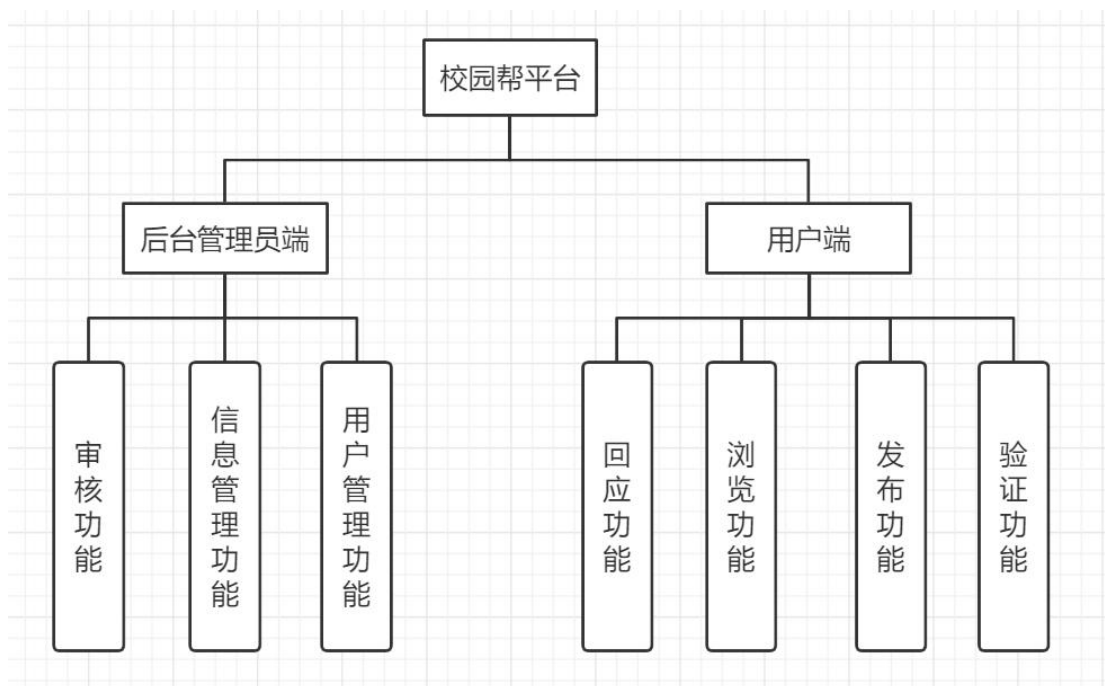


图 3

使用本平台及其架构的用户类如下：

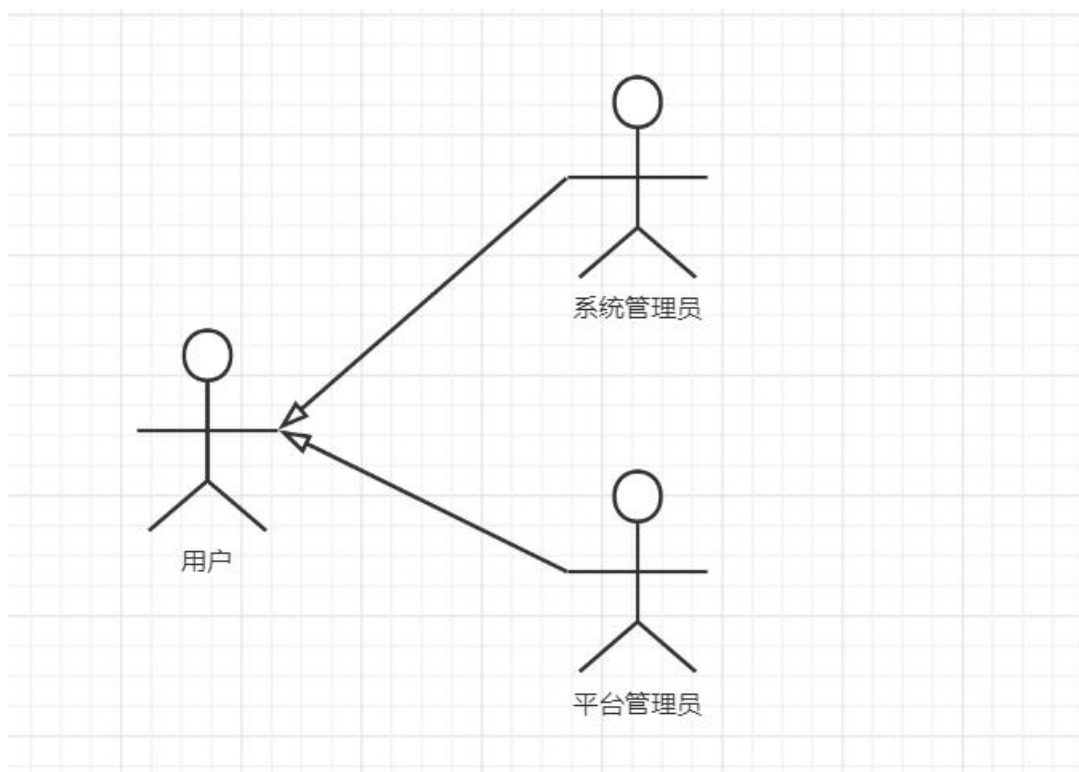


图 4

系统设计说明书

● 用户

通过用户信息交流平台，使用小程序中各个具体功能的操作用户，他们的一般的操作行为是：通过微信小程序登录平台，进入到平台的主界面，可以浏览到所有的发帖，点击发帖可以看到详细信息，也可以在发布界面进行信息的发布。同时在我的界面中可以通过点击“我的发布”来查看自己的历史信息发布情况，在“审核结果”中可以查看自己申请发帖的审核情况。

用户可以通过微信授权后直接登录到小程序中，同时在经过学生证验证后可以进行信息的交流，提高使用的方便性和信息的可靠性。

● 平台管理员

集中工作平台的管理员主要负责维护集中工作平台的应用基础框架，如用户人员的管理，信息发布的审核，信息的删除，回应信息的审核。

集中工作平台管理员要求对集中工作平台的架构和相关接入规范比较熟悉，对计算机应用系统的操作比较熟练。

● 系统管理员

应用系统开发人员指接入平台的项目开发团队技术人员，他们需要了解平台的相关架构、公用框架、模型、规范、接口，以确保实施的应用能够无缝接入到集中工作平台中。另外，开发人员需要将开发的功能定义到平台的功能管理中。

应用系统开发人员熟悉电脑操作和软件开发技术。

在系统机构设计和功能设计上，要求充分考虑用户类的使用特征，更好满足使用者的操作体验。

2.5 系统架构

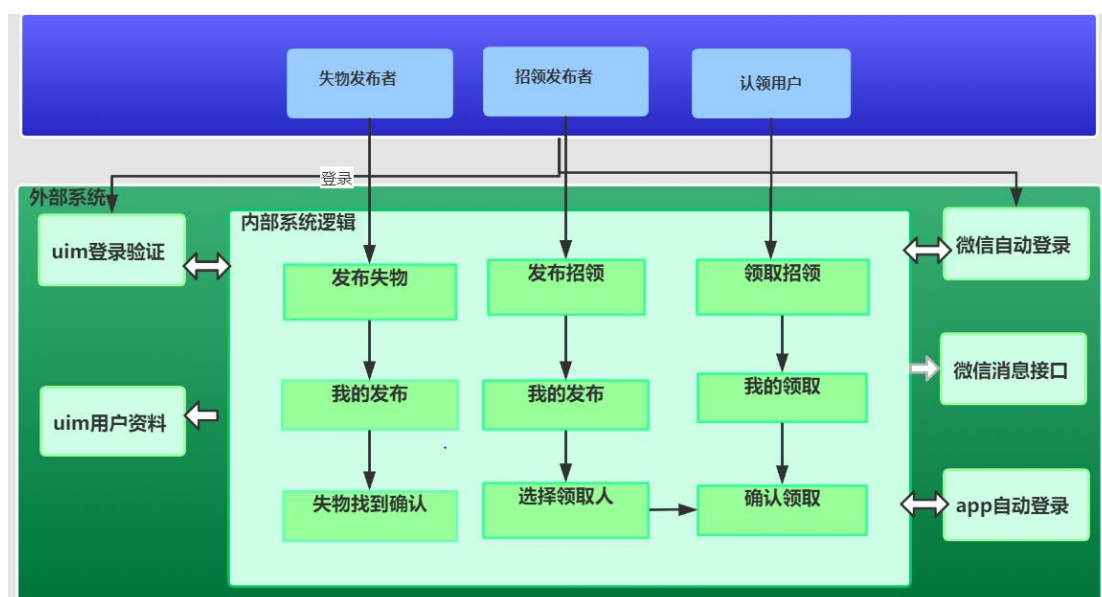


图 5

系统设计说明书

2.5.1 技术架构

系统搭建一般以 J2EE 作为开发基础、MySQL 作为数据库、Eclipse 为开发工具、以微信小程序为载体, Bomb 为后端, 实现内嵌于微信小程序平台的校园失物招领系统小程序设计. 失物招领平台系统包括丢失物品发布、拾取物品发布、查询统计信息、后台用户管理等功能。

系统设计说明书

2.5.2 功能架构

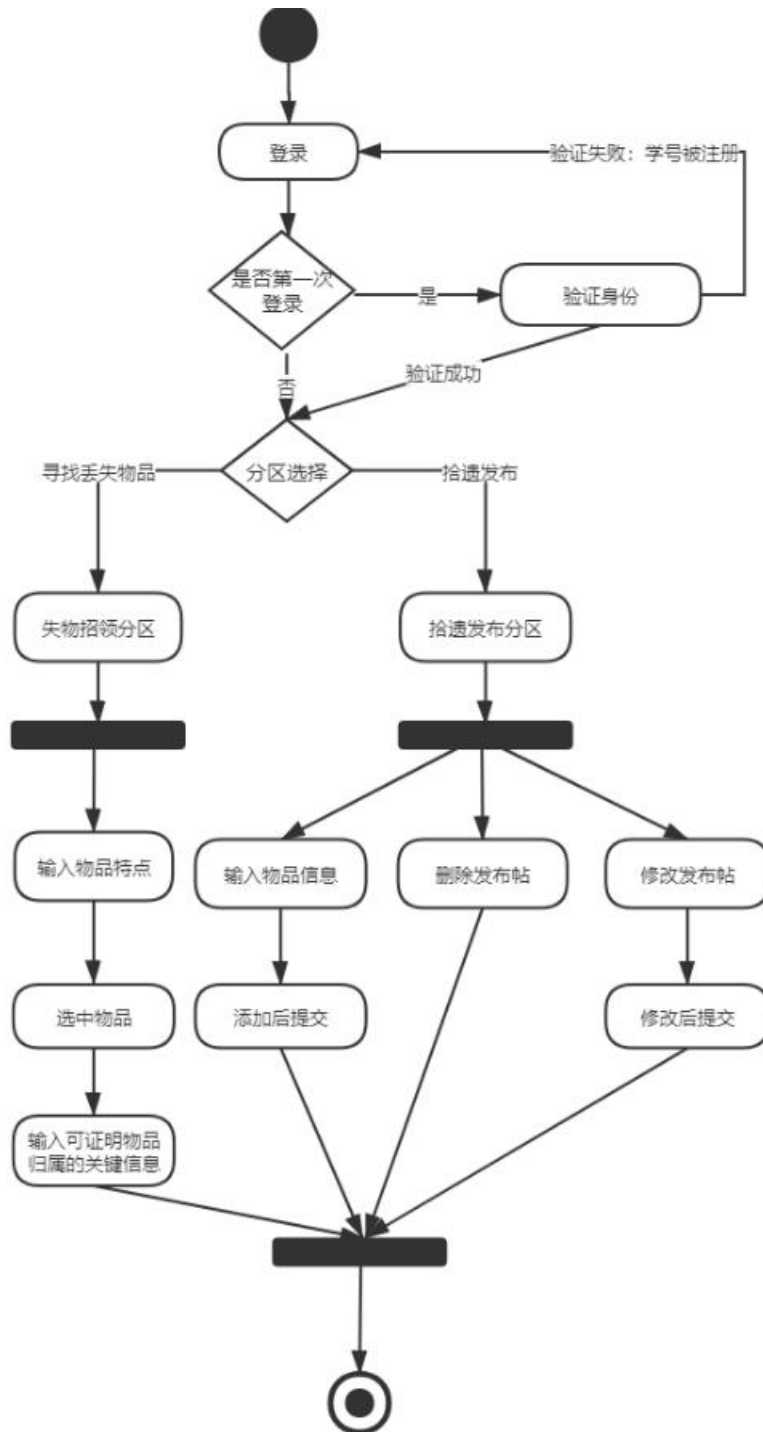


图 6

系统设计说明书

2.5.3 UI 架构



图 7

系统设计说明书



图 8

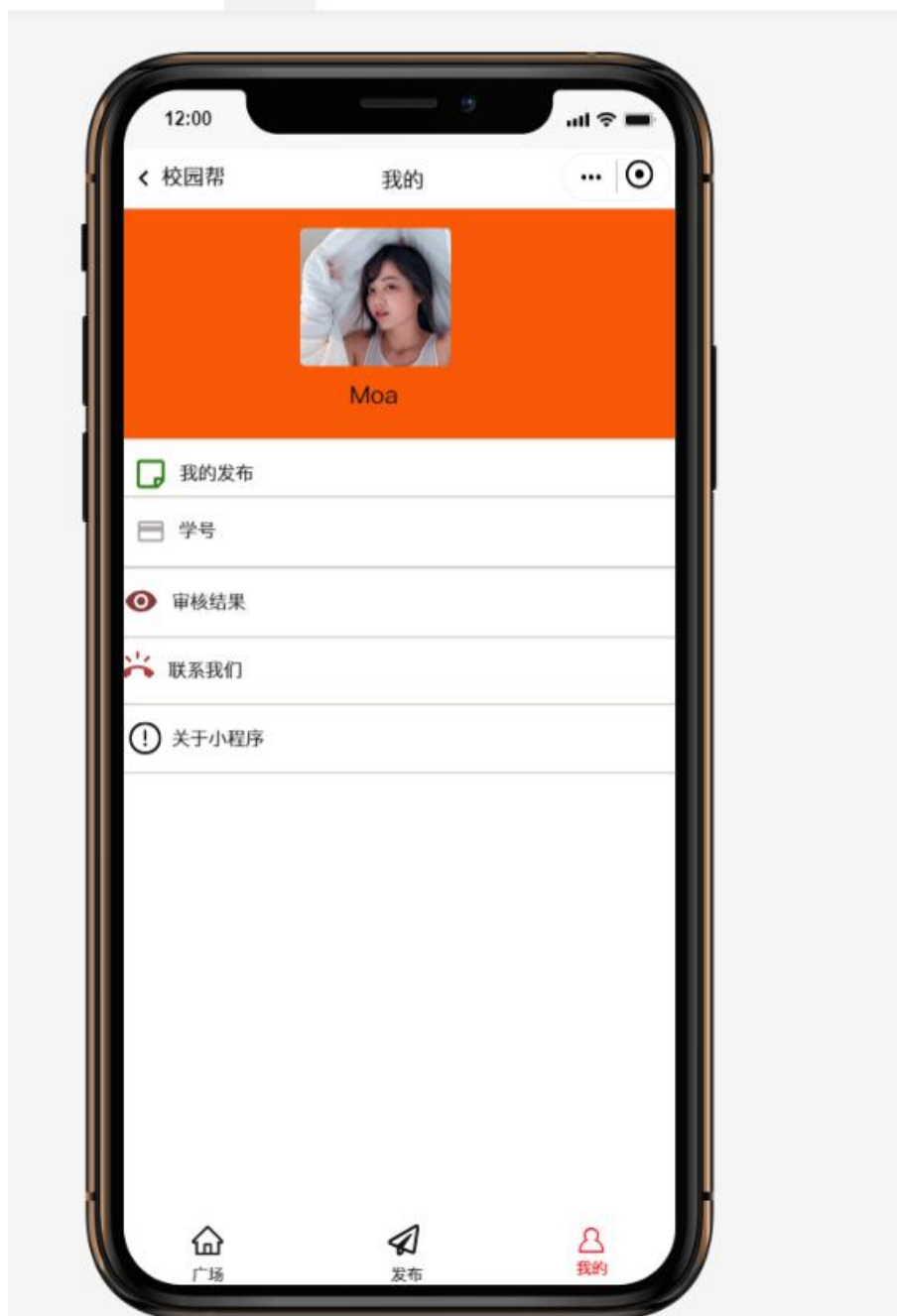


图 9

2.6 接口设计

1、硬件接口

客户端：

Android 兼容的手机

ARMEABI-v7a 兼容的 CPU

系统设计说明书

GPS 模块

网络连接支持

服务器端：

CPU：

建议 2.0Ghz 以上至少四个逻辑核心

内存：建议 8G 以上

带宽：建议 100Mbps

存储：建议 100G 以上

2、软件接口

服务器端：

Java 运行时版本 12 或以上

数据库：PostgreSQL 11 以上，带有 PostGIS 扩展

客户端：

Android 版本 7.1 或以上（API 26）

系统位置服务启用

2.7 系统环境

1. iOS 平台，包括 iOS9、iOS10、iOS11
2. Android 平台
3. 小程序 IDE

2.7.1 硬件环境

各类 android 手机或者平板，或者 IOS 手机或者平板

2.7.2 软件环境

android2.2 及以上版本，或者 IOS7.0 及以上版本

系统设计说明书

2.8 非功能特性设计

2.8.1 系统兼容性

机型和系统版本：小程序部分 api 依赖于系统底层实现，以及 JavaScript 标准库也与系统版本强关联

历史数据：小程序本地缓存历史数据与新版本的兼容

发布：最新微信客户端小程序基础库发布有个灰度过程，一般为 12 个小时，API 新老版本兼容

2.8.2 安全性

安全性：

- 1、恢复：本产品存在草稿箱功能，可存储所输入的内容，可保证信息安全。
- 2、权限控制：本产品针对用户设置权限设置，以防信息泄露。
- 3、最小权限原则：只申请必要的权限，避免成为攻击入口。
- 4、本产品用户可对其他用户进行举报信息反馈，使平台内容以及用户更加安全。

2.8.3 运行效率

支持终端数量：3000

支持并发请求数量：150

服务器数据区存储容量：100G

2.8.4 可扩展能力

可持续进行更新，未来可根据用户的意见为之增加新的功能。

3 系统公共模块设计

3.1 权限控制

每个集合可以拥有一种权限配置，权限配置的规则是作用在集合的每个记录上的。出于易用性和安全性的考虑，云开发为云数据库做了小程序深度整合，在小程序中创建的每个数据库记录都会带有该记录创建者（即小程序用户）的信息，以 `_openid` 字段保存用户

系统设计说明书

的 `openid` 在每个相应用户创建的记录中。因此，权限控制也相应围绕着一个用户是否应该拥有权限操作其他用户创建的数据展开。

以下按照权限级别从宽到紧排列如下：

- 仅创建者可写，所有人可读：数据只有创建者可写、所有人可读；比如文章。
- 仅创建者可读写：数据只有创建者可读写，其他用户不可读写；比如用私密相册。
- 仅管理端可写，所有人可读：该数据只有管理端可写，所有人可读；如商品信息。
- 仅管理端可读写：该数据只有管理端可读写；如后台用的不暴露的数据。

简而言之，管理端始终拥有读写所有数据的权限，小程序端始终不能写他人创建的数据，小程序端的记录的读写权限其实分为了“所有人可读，只有创建者可写”、“仅创建者可读写”、“所有人可读，仅管理端可写”、“所有人不可读，仅管理端可读写”。

3.2 日志处理模块

日志记录与异常处理规范

3.2.1 日志记录规范

日志记录规范规范日志设计规范主要目的是节省工作量，帮助对问题进行诊断。最终，终端用户可以获得更好的应用程序，并能从技术支持团队获得迅速的响应。

3.2.1.1 日志分类

3.2.1.1.1 系统业务运行日志

业务系统运行情况：启动时间，运行时长，结束时间、运行结果（成功、失败等），基本描述信息。相关的信息要保存到业务日志文件中。

BusinessLog：系统日志

系统设计说明书

3.2.1.1.2 系统调试日志

DebugLog，打印处理就可以了

在系统编码过程中，方便系统查找原因，输出相关的日志信息，方便查找问题，在系统业务运行过程中不记录相关问题。

在系统运行后不再输出信息，调试过程中要用 assert 检查内容，其日志信息可以输出到文件中。

3.2.1.2 日志级别分类

INFOX:用于在产品环境中（粒度较粗）帮助跟踪执行过程的上下文消息。（系统运行的情况，相关信息要保存到日志中

DEBUG:包含了非常广泛的上下文信息，用于问题诊断。可以在任何时候，任何地方添加，用于系统调试

WARNX:警告消息，说明系统中可能存在问题。例如，如果这个消息类别是有关安全性方面的。运行的方面如下：多种选择时，没有该选择项目 if...elseif...else 或者 select 等

ERROR:错误消息说明系统中出现了严重的问题。这种问题通常都是不可恢复的，需要人工进行干预。（系统能够 Catch 到的问题都是此类问题）

1、数据库连接失败、数据入库失败等、文件打开失败、数据参数读取失败等由于配置等原因导致系统无法正常运行

2、缺少系统运行所需的配置文件，参数文件等

系统设计说明书

3、系统没有按照设计内容运行，没有喜爱那个个内容系统无法运行

FATAL=4//异常，在程序正常运行时永远都不可能出现的状态。

1、无法申请内存（满载）

2、参数越界（超过定义内容）

告警类型：

```
#define CPLE_None0
```

```
#define CPLE_AppDefined1
```

```
#define CPLE_OutOfMemory2
```

```
#define CPLE_FileIO3
```

```
#define CPLE_OpenFailed4
```

```
#define CPLE_IllegalArg5
```

```
#define CPLE_NotSupported6
```

```
#define CPLE_AssertionFailed7
```

```
#define CPLE_NoWriteAccess8
```

```
#define CPLE_UserInterrupt9
```

```
#define CPLE_ObjectNull10
```

3.2.1.3 规范

1. 不要把 `system..out` 用于日志记录。

2. 被认为“已完成”或无隐错的代码，也应该能够根据配置输出日志（因为总有存在隐错的可能性，可能是通过修改引入的）。

3. 除非代码能够生成日志消息，并且它的日志输出能够被容易的配置，否则它不可用于生产环境。

系统设计说明书

4. 在重要代码段中需要详尽的日志记录输出。
5. 在用于生产环境前，需要修改和改进维护期间的日志记录陈述（例如，如果日志输出显得不明确）。
6. 日志消息应该分成不同优先级，并且调试消息应该指出一个构件的整个工作流程。
7. 明确日志消息优先级的选择，具有相同优先级的日志消息应该揭示一致级别的细节。
8. 从异常中提取有用信息
 1. 发生一个非预期的异常时：首先，必须记录异常及其堆栈跟踪状况。其次，应该使用一种用户界面友好的方式来标识错误页面，这对于终端用户和技术支持小组来说都是非常有帮助的。
 2. 使用户所报告的问题与特定的日志异常之间建立某种关联：为每个异常都记录一个唯一的 ID。这个 ID 可以告诉用户，也可以包含在终端用户所填写的问题报告表单中。让技术支持团队可以快速对问题作出响应。考虑到可读性的问题。
9. 确保日志记录语句中调用 `toString()` 方法的对象不会是空值，以免这些语句引起失败。
10. 通过相应的配置，用来显示生成日志输出的那个类、方法和行号（Log4j 提供此功能）。该设置在生产中应该被关闭，因为生成这类信息的代价非常高。
11. 记录日志时不要把 `toString()` 方法用在访问具有许多方法并生成大串的复杂对象上。（也可以给这种日志输出定义一个低优先级的输出级别）。

3.3 异常处理模块

3.3.1.规范

1. 当程序产生异常时,必须捕捉并处理异常、将异常记录到日志中(除非打算抛出异常)。
2. 不要处理能够避免的异常。
3. 在异常处理模块中提供适量的错误原因信息。
4. 不要从 try 区段中返回。
5. 不要从 try 区段中返回。
6. 不要从 try 区段中返回。
7. 将 try/catch 区段置于循环之外。
8. 不要将异常用于程序流程控制。
9. 通过抛出异常,获取构造函数中的错误信息。
10. 抛出异常之前先将对象恢复为有效状态。
11. 记录异常不要保存 `exception.getMessage()`,而要记录 `exception.toString()`。
12. 一个方法不应抛出太多类型的异常。说明:如果程序中需要分类处理,则将异常根据分类组织成继承关系。如果确实有很多异常类型首先考虑用异常描述来区别, `throws/exception` 子句标明的异常最好不要超过三个。
13. 不要定义 `Error` 和 `RuntimeException` 的子类,可以定义 `Exception` 的子类。

系统设计说明书

3.3.2 建议

1. 在捕获违例的时候，不使用 `Exception`, `RuntimeException`, `Throwable`，尽可能使用它们的子类。
2. 异常捕获尽量不要直接 `catch(Exception ex)`，应该把异常细分处理。
3. 在程序中使用异常处理还是使用错误返回码处理，根据是否有利于程序结构来确定，并且异常和错误码不应该混合使用，推荐使用异常。
4. 尽量减小 `try` 块的体积。
5. 尽量抛出异常，顶层的 `main()` 函数截获所有的异常，并且打印（或者记录在日志中）在屏幕上。

3.3.3 异常处理模块

- 1、在构件包中建立异常资源文件，针对错误码，包括显示在页面的提示和真正的错误提示，例如：

`abf_auth_0001=无效的用户名或者密码![用户{0}密码错误!]`

其中 `{0}` 表示变量，如果异常提示中有多个变量，依次为 `{0}`，`{1}`...`[]` 内的信息为真正的错误信息。如果没有，则和前面信息一致。显示给用户的信息为 `[]` 前的信息。

异常资源文件为构件包资源配置目录下 `resources/exception/exception.properties`

（1）错误码的构成规则

构件包名+”_”+四位数字：例如 `abf_auth_0001`

（2）实现一个写业务异常的运算逻辑

接口如下：

```
public static void setBusiException(String errCode, String logFlag, String...  
params) throws com.eos.system.exception.EOSException
```

第一个参数是资源文件中定义的错误码

第二个参数为是否写业务日志的标志，缺省为 “true”，其他值为不写

第三个参数开始，为异常资源中的变量，当异常资源定义中，对应错误码的信息汇

系统设计说明书

总有{0},{1},{2},则需要设置第三、四、五个参数

在该运算逻辑中,将实现如下处理逻辑:

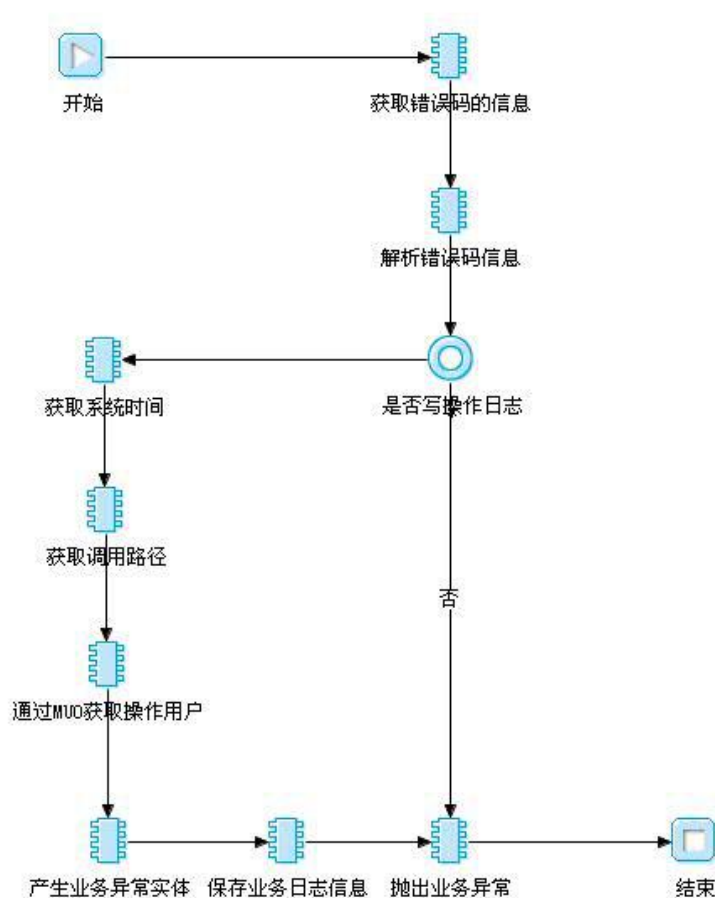


图10

2、业务逻辑中,当需要进行业务异常提示时,调用写业务异常的运算逻辑,由于运算逻辑会抛除异常,所以该运算逻辑将连接在一个结束图元前。

3、通过 EXT 提供一个显示业务异常的提示页面,当发生异常时,弹出业务异常提示窗口。

3.4 Session 处理

Session 将保存如下信息:

1) 用户基本信息:

```
<USERSESSION>
  <STUDENTID/>
  <NAME/>
  <WECHATID/>
```

系统设计说明书

```
<WECHATNAME/>
<WECHATAVATAR/>
<DATAIME/>
</USERSESSION>
```

2) 用户操作功能:

```
<OPERATIONSESSION>
  <Publish/>
  <browse/>
  <query/>
  <Respond/>
</OPERATIONSESSION>
```

3) 管理员基本信息:

```
<administratorsSESSION>
  <Account/>
  <Name/>
  <Passwd/>
  <Avator/>
  <Datetime/>
<administratorsSESSION/>
```

4) 管理员操作功能:

```
<MANAGE OPERATIONS>
  <DELETE USER/>
  <TO EXAMINE/>
  <BROWSE/>
  <QUERY/>
<MANAGE OPERATIONS/>
```

3.5 UI 框架设计例子

微信小程序设计 ui 框架:

weui-wxss 是微信官方提供的 ui 框架,用起来也很方便(copy)。

下载地址: <https://github.com/weui/weui-wxss>

第一步: 需要把下载的内容 copy 到项目中, 主要是 dist 目录下的 style 目录下的 weui.wxss 文件。

系统设计说明书

第二步：在全局的 app.wxss 文件中引用，注意目录是相对路径

```
@import "style/weui.wxss";
```

第三步：加 class 样式

```
class="page">
class="page__bd">
name="msgTemp">
class="weui-panel__bd">
url="../detail/detail?id={{id}}"
class="weui-media-box weui-media-box_appmsg"
hover-class="weui-cell_active">
class="weui-media-box__hd weui-media-box__hd_in-appmsg">
class="weui-media-box__thumb" src="{{src}}" style="width: 60px; height:
60px;"/>
class="weui-media-box__bd weui-media-box__bd_in-appmsg">
class="weui-media-box__title">{{title}}
class="weui-media-box__desc">{{time}}
class="weui-panel weui-panel_access">
class="weui-panel__hd">文章列表
wx:for-items="{{msgList}}" wx:key="{{item.id}}">
class="kind-list__item">
is="msgTemp" data="{{...item}}"/>
class="weui-panel__ft">
class="weui-cell weui-cell_access weui-cell_link">
class="weui-cell__bd">查看更多
class="weui-cell__ft weui-cell__ft_in-access">
class="page__ft">
```

这个步骤很关键，怎么找到合适的文档呢？weui-wxss 是没有文档的，但是它有例子。在下载的内容中，dist 目录下的 example 目录下全部都是例子。我们需要做的是将 dist 目录导入 IDE 中，然后找到合适的 UI。

4 应用基础框架子系统设计

4.1 系统框架

4.1.2 概述

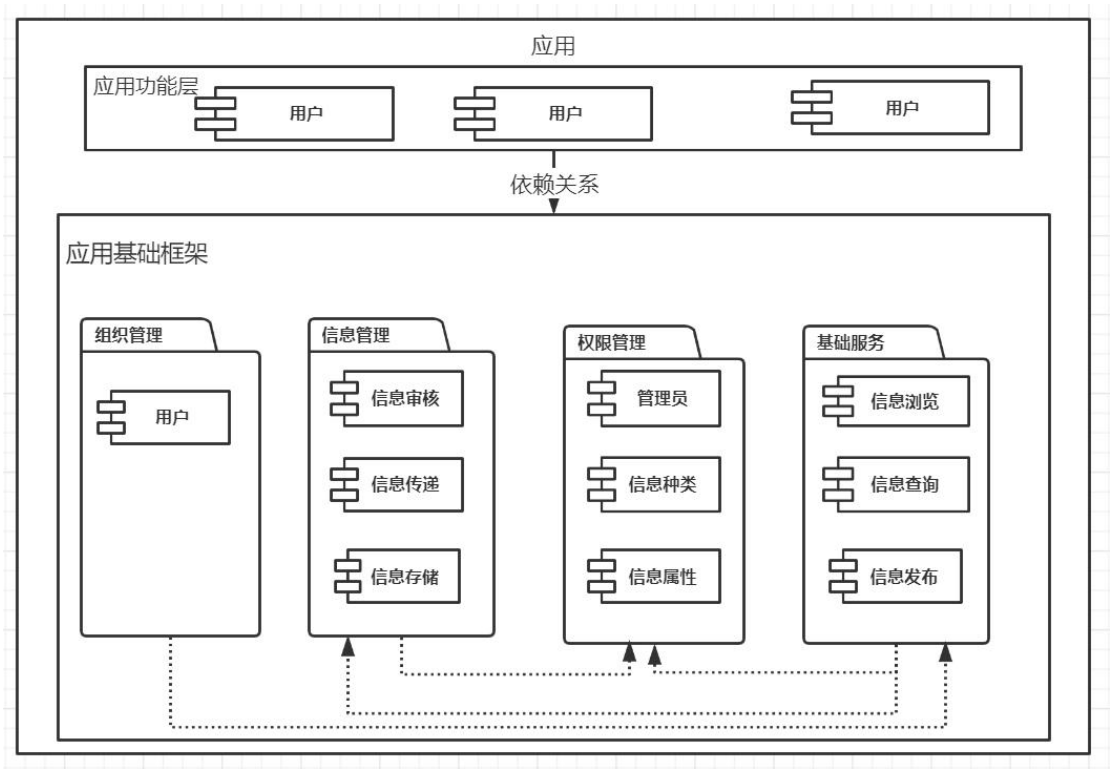


图 11

该框架为信息发布平台提供基础的支撑，同时也为未来接入数据库的用户提供基础权限接入，使用应用基础框架的主要场景和作用包括：

- 1、该框架可以实现对数据的统一管理，方便管理员统一管理。
- 2、方便数据统一通过管理员进行调用、导出。
- 3、数据在更新时调用相关数据库，保证数据的一致性。
- 4、通过一位系统管理员对用户的一些数据进行修正。
- 5、管理员对用户和信息作出一系列管理操作。

4.1.2 用例描述

27

系统设计说明书

		间搜索		
用户	举报	在‘帖子详细’界面下点击举报按钮	对发布违规信息用户进行举报	低
管理员	对用户进行管理	点击用户管理按钮	对用户进行管理	高
管理员	删除用户	点击用户信息右侧的‘×’号	对违规用户进行删除	高
管理员	查找用户	在搜索栏出对用户进行搜索	搜索用户	高
管理员	账号密码登录	初始界面	登录管理员账号	高
管理员	头像上传	登录后点击头像框	使用管理员账号进行登录	低
管理员	发帖审核	点击帖子管理界面下的失物帖	对提交的失物帖进行审核	高
管理员	找回审核	点击帖子管理界面下的寻回帖	对提交的找回帖进行审核	高
管理员	驳回\通过	帖子管理界面对文件进行审核	对提交的帖子进行通过和驳回的操作	高
管理员	对现有发布信息进行审核	登录后点击‘帖子审核按钮’	对帖子现有信息进行审核	高

4.1.2.1 权限管理的用例描述

权限管理部分主要处理与权限管理控制相关的数据维护的用例，如下图：

系统设计说明书

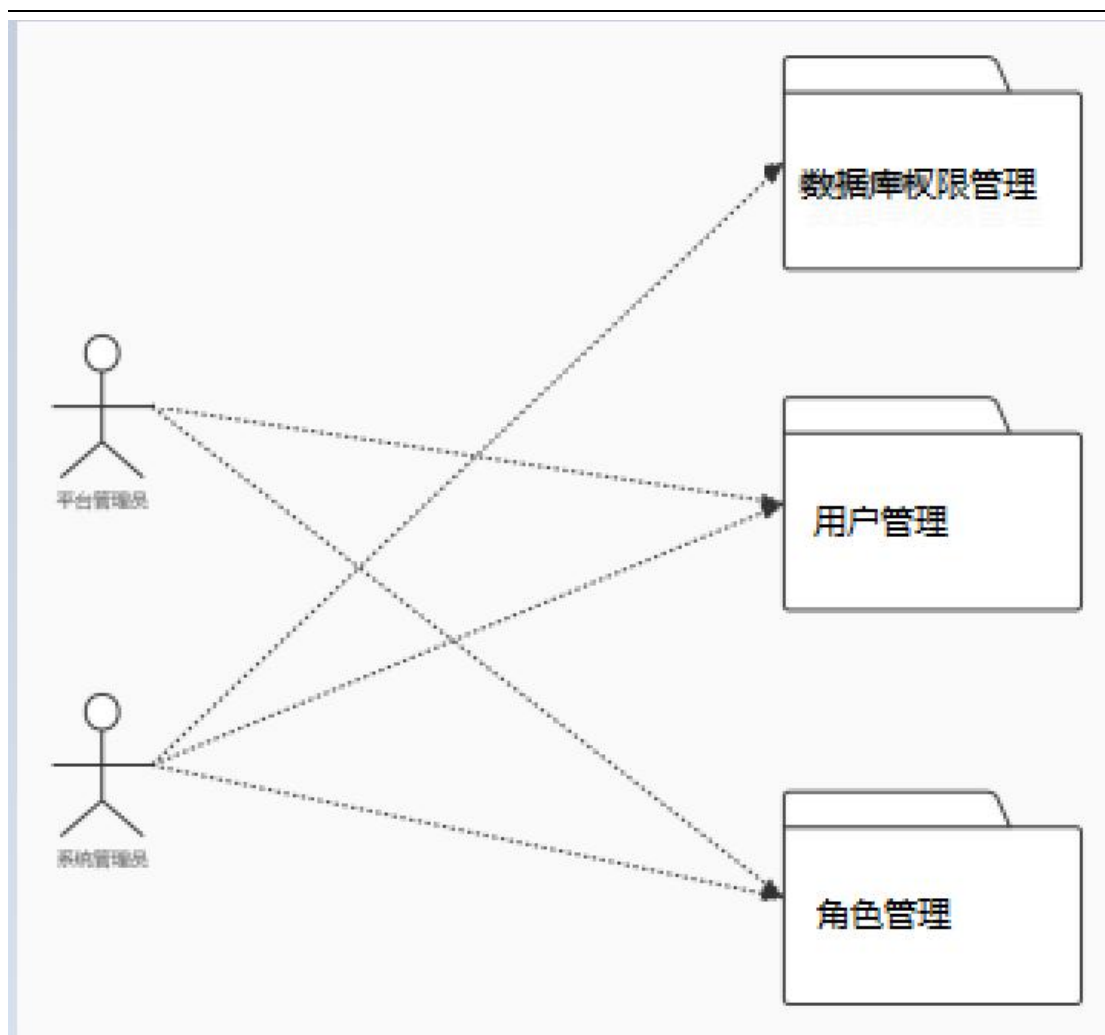


图 13

(2) 用例描述

系统设计说明书

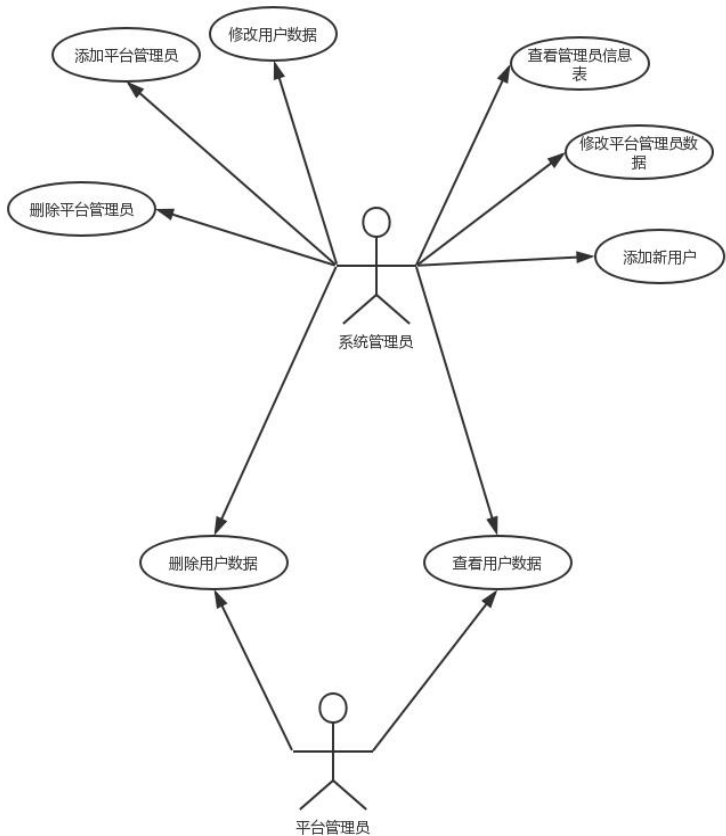


图 14

参与者	用例	触发方式	需求特性描述	优先级
平台管理员	查看用户数据	进入用户管理页面	对用户数据进行查看	高
平台管理员	删除用户数据	在用户管理页面时, 对用户违规用户数据进行删除。	对违规用户数据删除	高
系统管理员	删除平台管理员	数据库操作	删除在现实中不在对用户负责的管理员	高
系统管理员	修改平台管理员数据	数据库操作	对忘记密码的管理员执行修改密码的操作	高
系统管理员	添加平台管理员数据	数据库操作	对平台管理员进行添加	高
系统管理员	添加新用户	数据库操作	向用户表中添加一位新用户	高
系统管理员	查看用户数据	数据库操作	对用户数据进行查看	高
系统管理员	删除用户数据	数据库操作	对违规用户数据删除	高

系统设计说明书

系统管理员	修改用户数据	数据库操作	对用户部分错误信息进行修改	高
系统管理员	添加新用户	数据库操作	对被误封的用户重新添加回数据库	高

4.1.3 组织机构管理的用例描述

角色管理

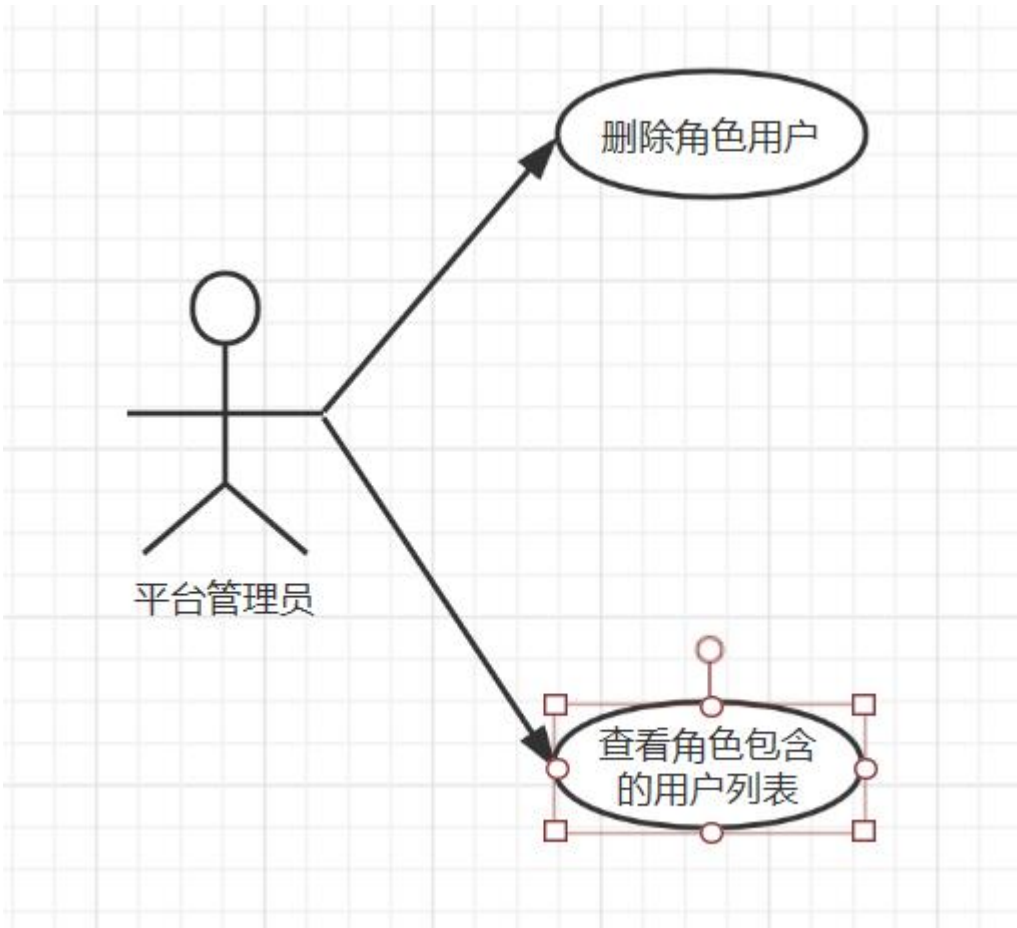


图 15

参与者	用例	触发方式	需求特性描述	优先级
平台管理员	删除角色用户	执行删除用户操作时	删除数据库中对应的数据	高

系统设计说明书

平台管理员	查看角色包含的用户列表	管理员查看用户数据	对数据进行查找	高
-------	-------------	-----------	---------	---

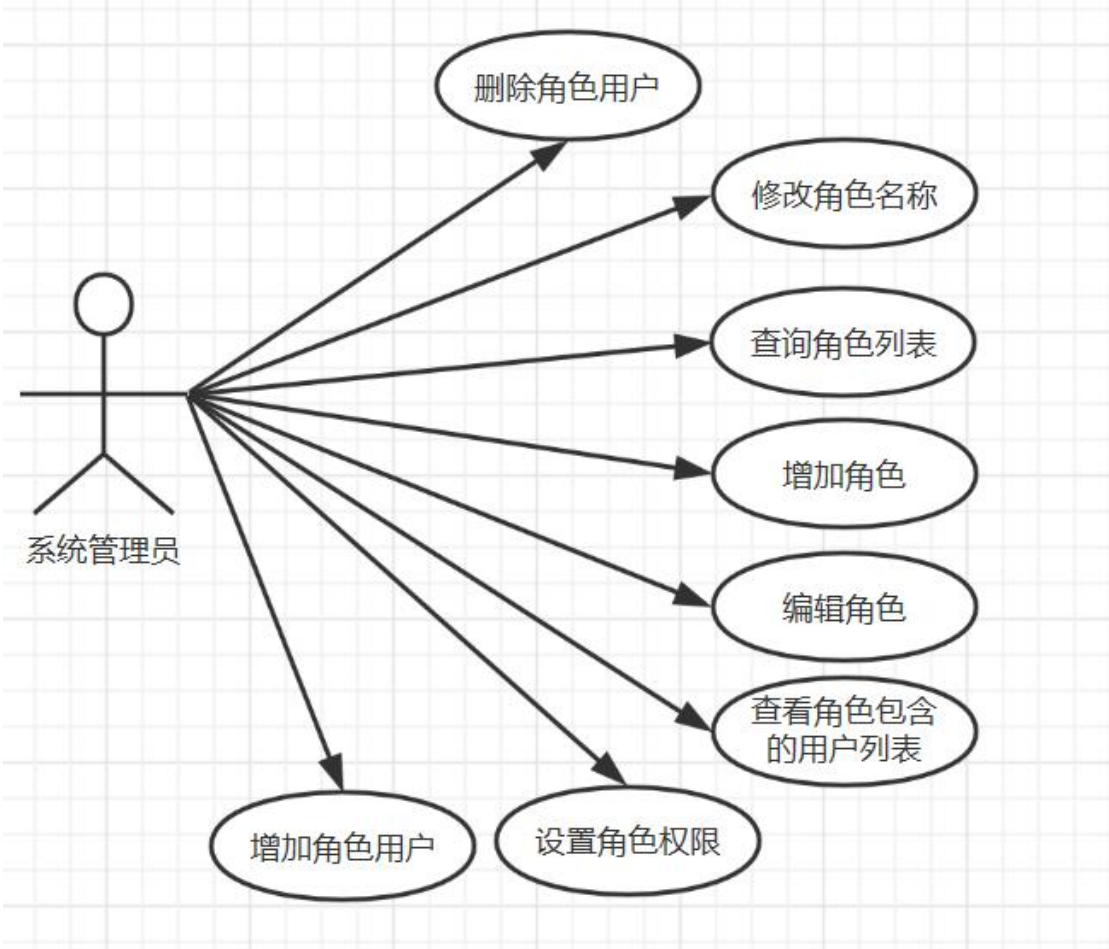


图 16

参与者	用例	触发方式	需求特性描述	优先级
系统管理员	删除角色用户	数据库操作	对数据进行删除	高
系统管理员	查看角色包含的用户列表	数据库操作	对角色中包含的用户进行查看	高
系统管理员	修改角色名称	数据库操作	对角色名进行修改	高
系统管理员	增加角色用户	数据库操作	添加某个角色下的用户	高
系统管理员	设置角色权限	数据库操作	对角色权限进行设置	高

系统设计说明书

系统管理员	编辑角色	数据库操作	对角色进行编辑	高
系统管理员	查询角色列表	数据库操作	查询数据库中有哪些角色	高
系统管理员	增加角色	数据库操作	向数据库中增加角色	高

(2) 用户管理

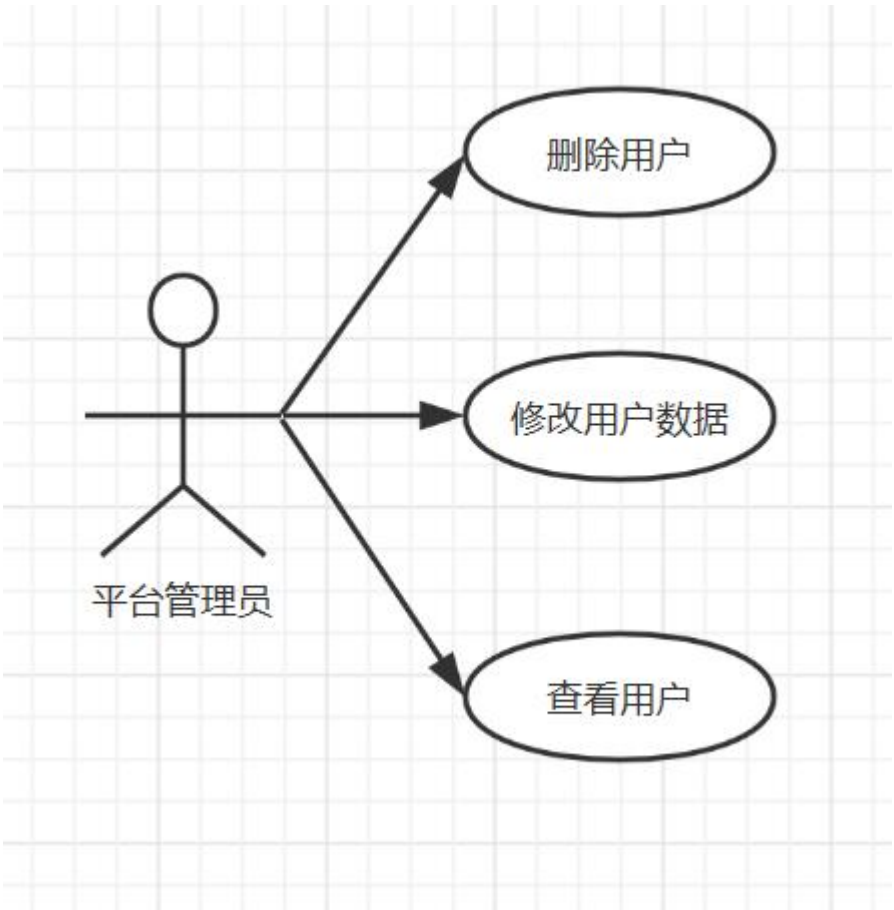


图 17

参与者	用例	触发方式	需求特性描述	优先级
平台管理员	删除用户	执行删除用户操作时	删除数据库中指定用户	高
平台管理员	修改用户数据	执行部分用户操作时	修改用户数据	高
平台管理员	查看用户	管理员查看用户数据	对数据进行查找	高

系统设计说明书

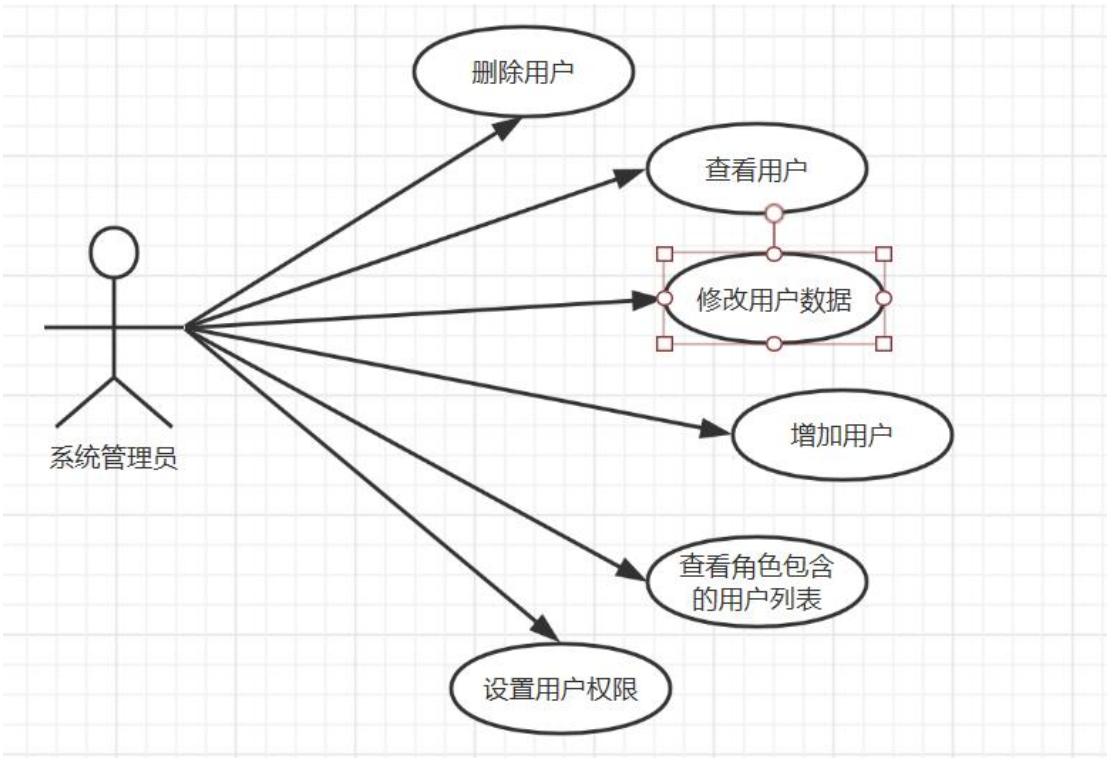


图 18

参与者	用例	触发方式	需求特性描述	优先级
系统管理员	删除用户	数据库操作	对数据库中用户执行删除操作	高
系统管理员	查看用户	数据库操作	对角色中包含的用户进行查看	高
系统管理员	修改用户数据	数据库操作	对用户信息进行修改	高
系统管理员	增加用户	数据库操作	新增用户	高
系统管理员	设置用户权限	数据库操作	对用户权限进行设置	高
系统管理员	增加用户	数据库操作	增加用户	高

(3) 数据库权限管理

系统设计说明书

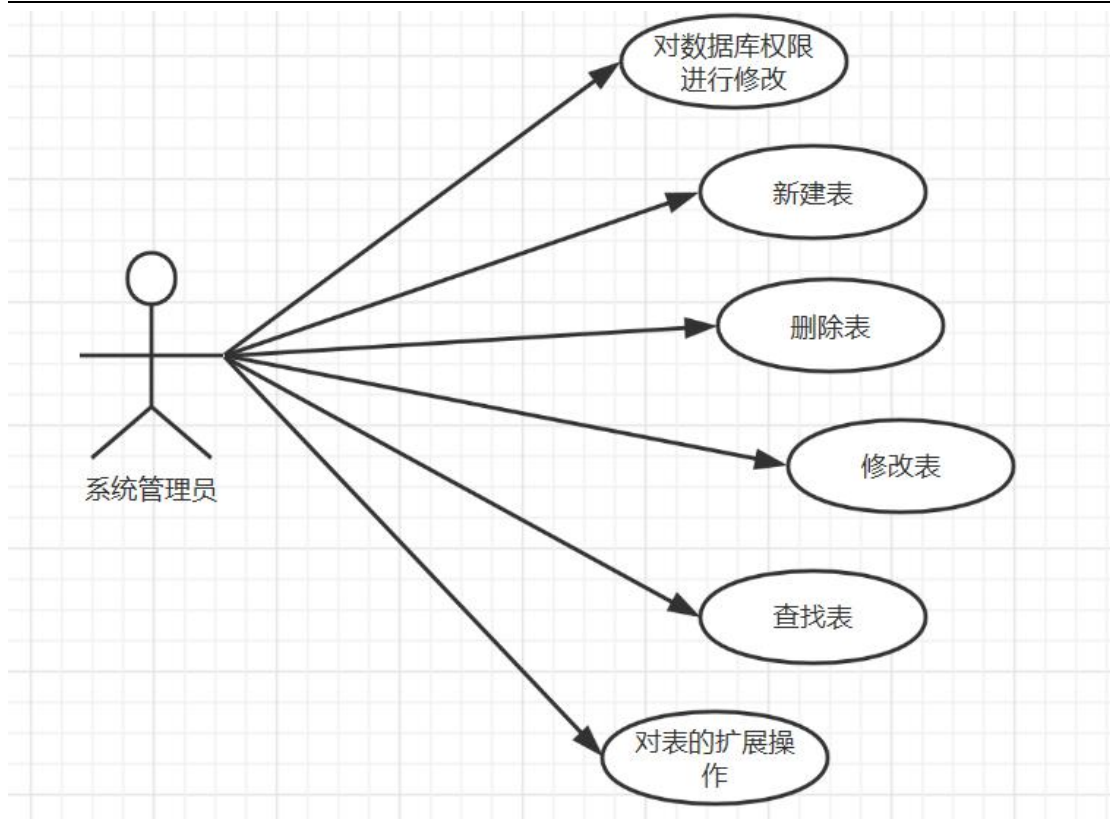


图 19

参与者	用例	触发方式	需求特性描述	优先级
系统管理员	对数据库权限进行修改	数据库操作	修改数据库权限	高
系统管理员	新建表	数据库操作	在数据库中建立新表	高
系统管理员	删除表	数据库操作	对表进行删除	高
系统管理员	修改表	数据库操作	对表结果或数据进行修改	高
系统管理员	查找表	数据库操作	对数据库中的表进行查找	高
系统管理员	对表的扩展操作	数据库操作	如建立索引等	高

系统设计说明书

4.1.4 实体关系描述（E-R 图）

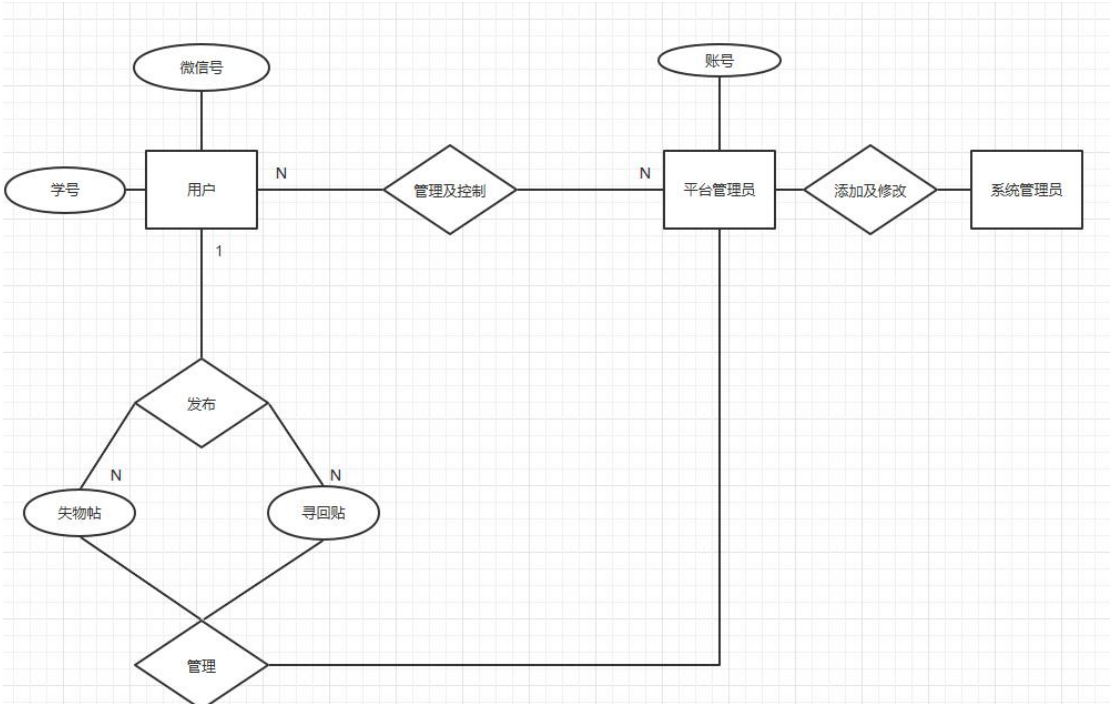


图 20

4.2 构件包设计

4.2.1 构件包列表

编码	包名称	包版本	包功能说明	依赖包
1	组织管理	1.0	对用户的管理	
2	信息管理	1.0	对信息的审核、浏览和存储管理	
3	基础服务	1.0	小程序的基础服务功能	
4	权限管理	1.0	对管理人员和数据库的管理	

4.2.2 构件包关系图

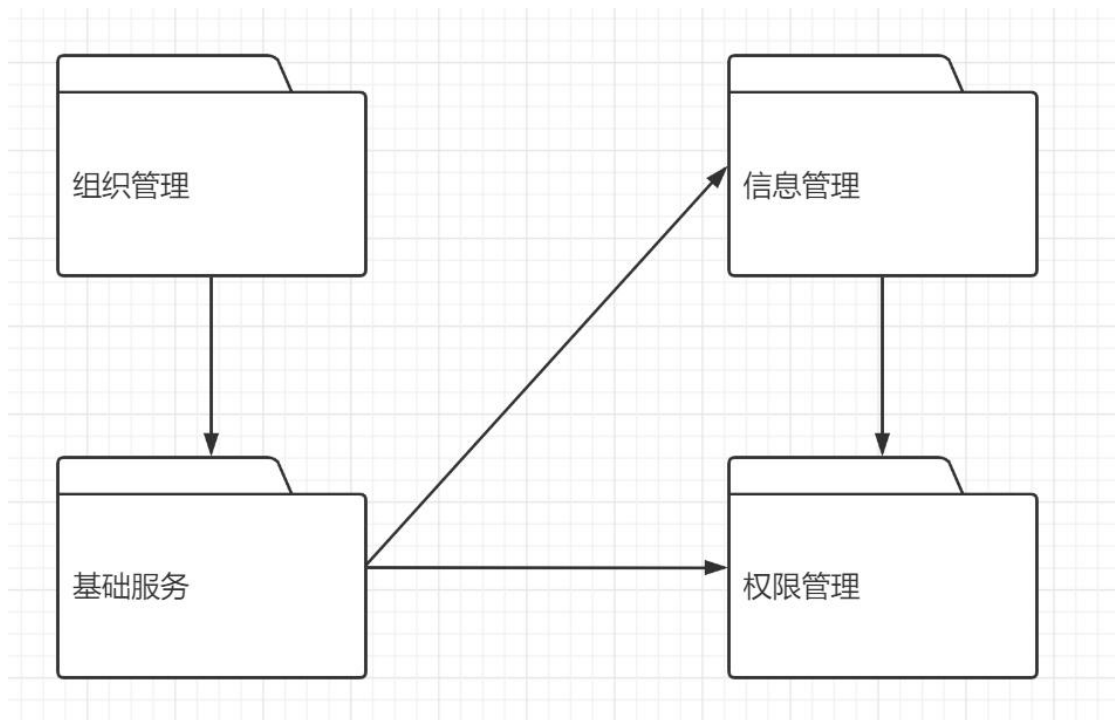


图 21

4.2.3 基础服务构件包

4.2.3.1 概述

软件的基础服务功能，有发布帖子信息，浏览帖子信息，对帖子信息进行查询等服务功能。

4.2.3.2 流程设计

4.2.3.2.1 信息发布流程

(1) 流程图

系统设计说明书

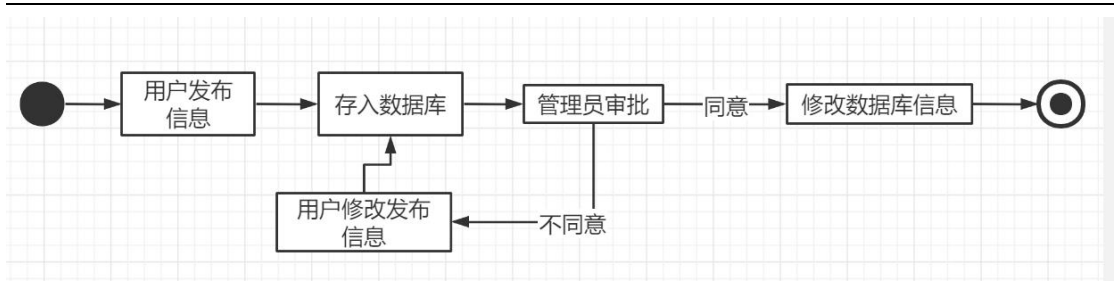


图 22

(2) 活动描述

活动名称	活动类型	参与者	会签规则	时限要求	触发事件	调用功能	其他特性
用户发布信息	人工活动	用户	无	无	无	信息发布	无
存入数据库	自动活动	数据库+软件系统	无	3 秒内	用户发布信息	信息存储	无
用户修改发布信息	人工活动	用户	无	无	修改信息	信息发布	无
管理员审批	人工活动	管理员	无	1 天内	通知管理员处理	信息审核	无
修改数据库信息	自动活动	数据库+软件系统	无	无	管理员审核完成	信息修改	无

4.2.3.2.1 信息浏览流程

(1) 流程图

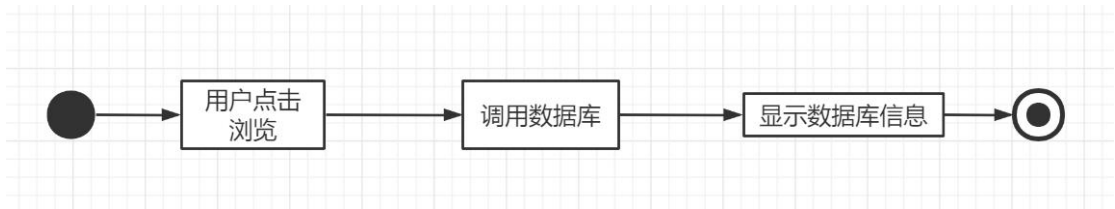


图 23

(2) 活动描述

活动名称	活动类型	参与者	会签规则	时限要求	触发事件	调用功能	其他特性
------	------	-----	------	------	------	------	------

系统设计说明书

用户点击浏览	人工活动	用户	无	无	无	使用平台	无
调用数据库	自动活动	数据库+软件系统	无	3秒内	用户使用平台	信息提取	无
显示数据库信息	自动活动	数据库+软件系统	无	无	数据库操作	显示信息	无

4.2.3.3 功能列表

功能编号	功能说明	增加时间	需求编号	备注
1	信息查找			按照特征字来查找信息。
2	信息过滤			通过选择时间来过滤信息
3	信息发布			用户发布信息
4	信息浏览			浏览所有的信息

4.2.3.4 功能设计

4.2.3.4.1 信息查找

(1) 功能描述

功能编号	1	对应需求编号	
功能名称	信息查找	优先级	2
设计者		最后更改者	
测试人员		测试日期	
使用对象	用户	调用方式	浏览界面输入
相关数据实体	帖子信息关键字		
相关页面	Browse interface	信息浏览界面	

(2) 功能页面

页面编号	2	页面名称	Browse interface
页面隐含规则说明			
帖子信息排列显示，同时对发布时间可以进行选择。			
隐藏数据项：发布时间、微信名，微信头像，帖子图片，帖子题目			
页面操作描述			
操作名称	描述		
查找	通过输入特征字进行信息查找		
浏览	直接在界面中对信息进行浏览		
过滤	透过选择时间来对信息进行过滤显示		

系统设计说明书

业务规则
帖子信息唯一，不能出现同样信息出现。
例外以及相应处理

- (3) 处理流程
- (a) 信息查找流程

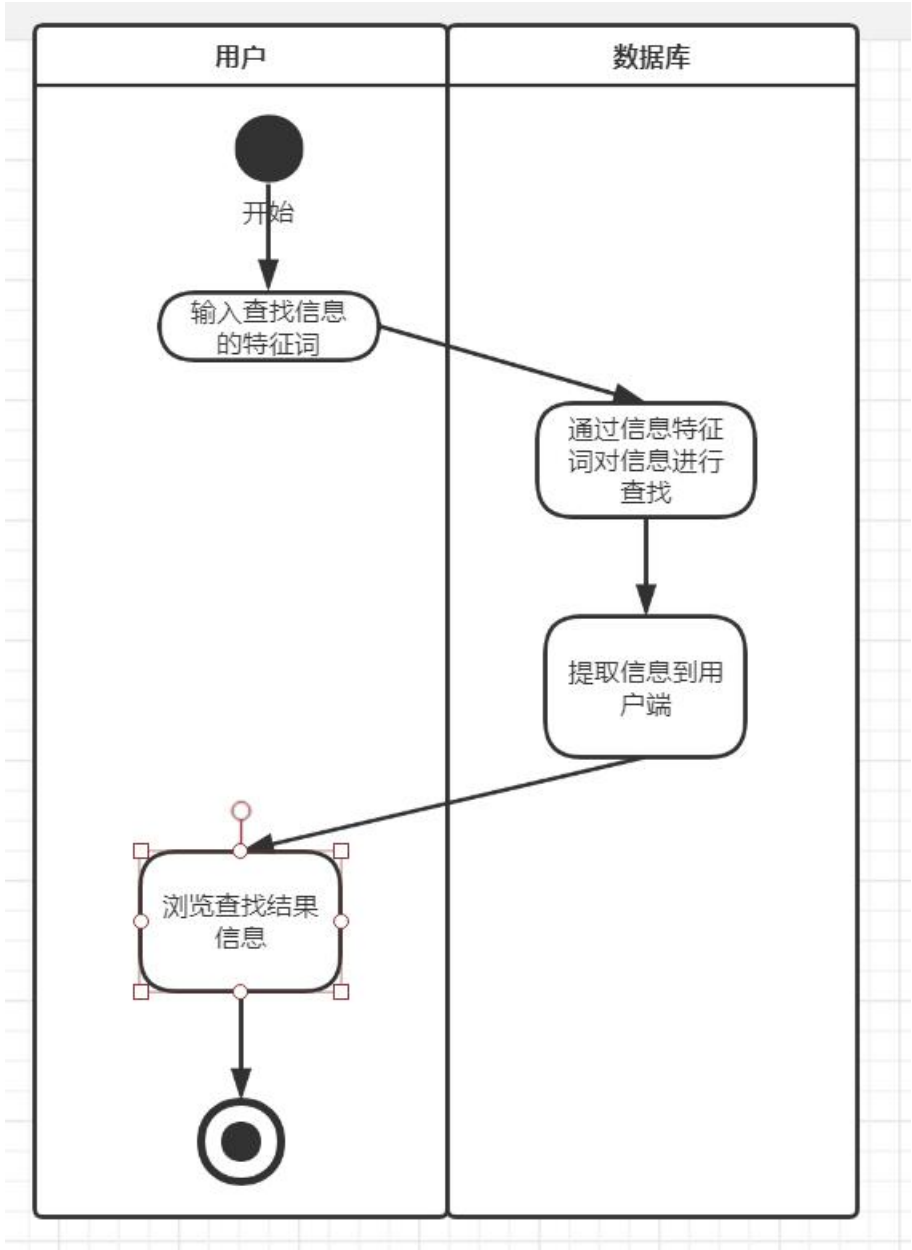


图 24

- (4) 用户界面

系统设计说明书

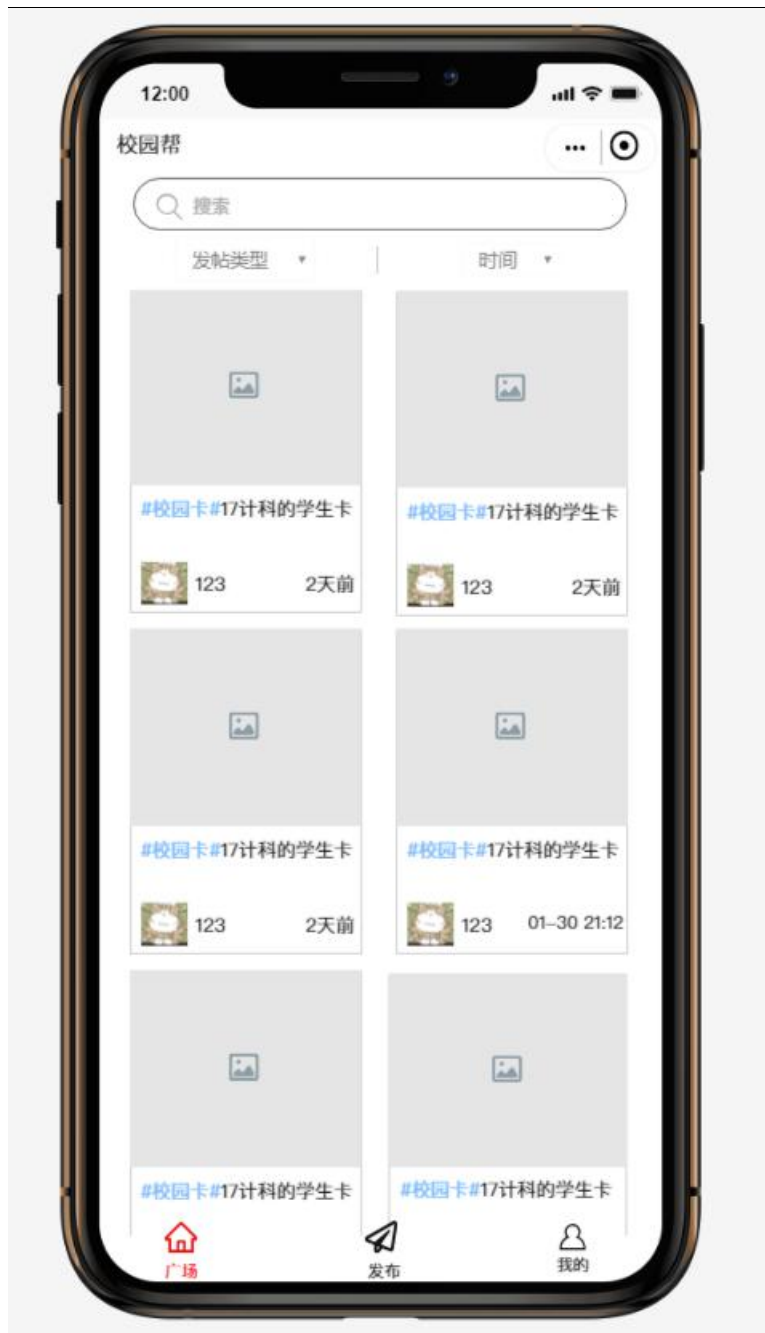


图 25

4.2.3.5 构件包的服务接口

接口类型分为五种：

业务服务：提供外部可以调用的业务服务接口，需要确定是否可以通过 webservice 调用

BL 方法类：提供外部可以调用的 BL 方法接口

JAVA 程序类：提供外部可以使用的 JAVA 程序

系统设计说明书

存储过程类：提供外部可以调用的存储过程接口

页面类：提供外部系统可以直接使用的页面功能（例如：工具在使用中需要外部配置参数，这些参数的维护页面就是页面类接口）

接口名称就是实际的包中具体的接口名称，对于页面类接口定义出进入页面的具体入口名称（可以是 JSP 和展现自动机名称）

（列举出业务模块需要外部暴露的接口，以及包之间需要调用的接口）

编码	接口名称	接口类型	功能详细描述	备注
1	用户操作员	页面类	用户进行页面操作	

4.2.3.5 构件包的引用接口

编码	接口名称	接口类型	功能详细描述	备注
1	用户操作员	页面类	用户进行页面操作	