

**Wink概要设计规约**

软件工程课程项目



**指导教师：**

**杜庆峰**

**小组成员：**

**1752026 李 航**

**1652977 王星洲**

**1650350 乔 宇**

**1753499 潘小逸**

目录

[修订历史 2](#_Toc29344744)

[1. 引言 3](#_Toc29344745)

[1.1 目的 3](#_Toc29344746)

[1.2 参考资料 3](#_Toc29344747)

[1.3 假定和约束 3](#_Toc29344748)

[1.4 定义 3](#_Toc29344749)

[2. 概要设计 4](#_Toc29344750)

[2.1 设计总体框架 4](#_Toc29344751)

[2.2 系统软件结构设计 5](#_Toc29344752)

[2.3 接口设计 5](#_Toc29344753)

[2.3.1 服务器IP和端口 5](#_Toc29344754)

[2.3.2 类之间接口详情 6](#_Toc29344755)

[2.3.3数据库接口详情 6](#_Toc29344756)

[2.3.4 系统与外部系统之间的接口 14](#_Toc29344757)

[2.4 界面设计 14](#_Toc29344758)

[2.4.1新闻详情页面 14](#_Toc29344759)

[2.4.2频道管理页面 15](#_Toc29344760)

[2.4.3新闻中心页面 16](#_Toc29344761)

[2.4.4评论/收藏/取消收藏新闻页面 17](#_Toc29344762)

[2.4.5新闻相关评论页面 18](#_Toc29344763)

[2.4.6动态搜索 19](#_Toc29344764)

[2.4.7新闻搜索页面 20](#_Toc29344765)

[2.4.8评论动态 21](#_Toc29344766)

[2.4.9动态详情页面 22](#_Toc29344767)

[2.4.10动态页面 23](#_Toc29344768)

[2.4.11关注页面 24](#_Toc29344769)

[2.4.12个人信息页面 25](#_Toc29344770)

[2.4.13用户中心页面 26](#_Toc29344771)

[2.4.14登陆注册页面 27](#_Toc29344772)

[2.4.15收藏页面 28](#_Toc29344773)

[2.4.16粉丝页面 29](#_Toc29344774)

[2.5 数据库设计 30](#_Toc29344775)

[2.5.1实体关系 32](#_Toc29344776)

[2.5.2实体属性 32](#_Toc29344777)

[2.6 系统出错处理设置 34](#_Toc29344778)

# 修订历史

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 编写日期 | SEPG | 版本 | 说明 | 作者 | 评审时间 | 评审人员 | 批准日期 | 签字人员 |
| 2019/11/12 |  | 1.0 | 初稿 | 李航 |  |  |  |  |
| 2019/11/19 |  | 1.2 | 增加数据库设计部分 | 乔宇 |  |  |  |  |
| 2019/11/26 |  | 1.3 | 数据库设计中增加表的新属性 | 乔宇 |  |  |  |  |

# 1. 引言

## 1.1 目的

本文档旨在将软件系统功能进一步细化，将原本复杂的功能拆分成不同的模块，并逐步开发健壮的系统结构，使软件系统需求能够很好的转化为未来设计，并进一步提高软件各方面性能。本文档依据高内聚低耦合的原则，将结构分解为不同的模块，各模块分别负责自己的功能，且模块与模块之间，模块与环境之间的调用关系尽可能简单。同时，在进行概要设计分析的过程中，发现并解决了系统中各个模块之间功能重合、冗余和功能短缺的部分，为详细设计和代码编写提供了保障。

本文档的目标读者为本软件的开发人员和其他相关人员。

## 1.2 参考资料

## 1.3 假定和约束

发布时间：2019年12月27日发布beta释放版

android版本要求：android SDK 16及以上

开发条件：成员自备电脑与移动应用，配置开发环境

经费支持：无经费支持

设计成本：自行承担设计成本

开发效率：使用SCRUM敏捷开发模型

## 1.4 定义

下面列出一些在本APP中可能出现的有特定含义的词汇

|  |  |
| --- | --- |
| 词汇 | 说明 |
| 用户ID | 作为识别用户的凭证 |
| 用户昵称 | 唯一的，能标识用户，用户用昵称和密码登录 |
| 频道 | 新闻界面不同类型的标签称为频道 |
| 新闻评论 | 不仅作为对于这个新闻的评论，同时作为一条动态自动发表 |
| 发表动态 | 发表一条不带新闻的动态 |
| 动态评论 | 区别于新闻评论，动态评论是仅仅针对动态的评论 |
| 关注 | 关注一个用户即代表成为对方的粉丝，用户可以进行关注或者取消关注的操作 |
| 粉丝 | 取消关注对方意味着你从对方的粉丝列表里消失，但是你的评论依旧在系统里，你重新关注他，你的评论会重新出现；但是当你被拉黑时，你的评论会消失，即使你重新关注了对方，你的评论也不会重新出现 |

# 2. 概要设计

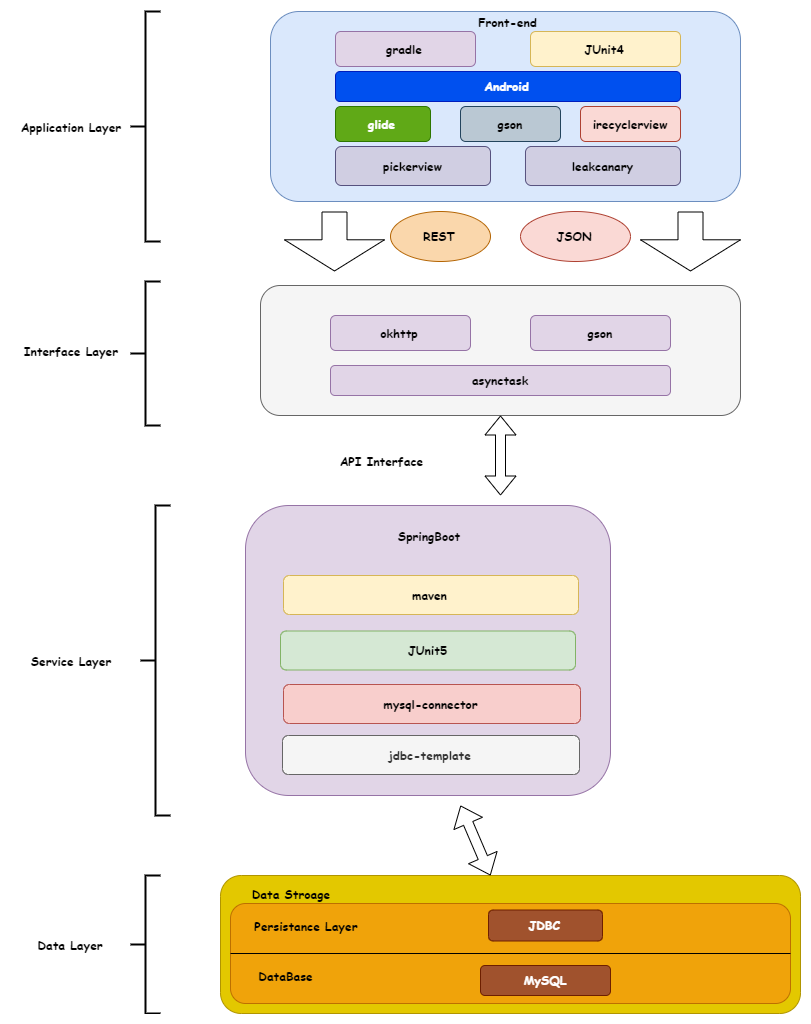
## 2.1 设计总体框架

应用层(前端)由于本应用是新闻类应用，用户希望随时随地使用的原因而采用了移动端开发，这里我们选择了安卓平台开发。采用了JUnit4和leakcanary进行测试以及内存检测，同时我们使用了irecyclerview，pickerview等用于组件设计。

接口层采用了okhttp发送http请求，用gson进行json与类之间的转换，asynctask用来实现异步访问。前后端接口设计采用了RESTful API

服务层(后端)我们采用了SpringBoot框架，用maven安装了依赖，并进行依赖注入，采用Junit5进行单元测试并用mysql-connector和jdbc-template实现与MySQL数据库的连接。

数据层我们选择了MySQL数据库，这是一个关系型数据库。并实现了它与后端的连接。具体架构图如下：



## 2.2 系统软件结构设计

## 2.3 接口设计

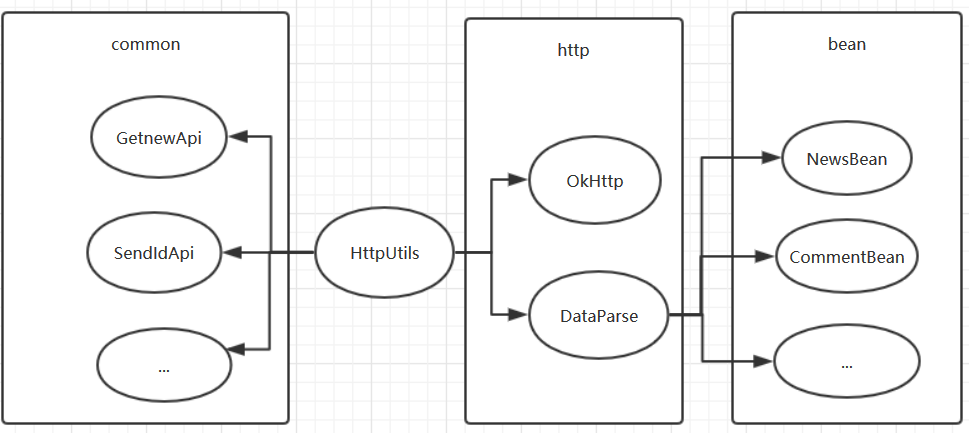
### 2.3.1 服务器IP和端口

本系统中的服务器IP和端口为47.100.79.111:8088

### 2.3.2 类之间接口详情

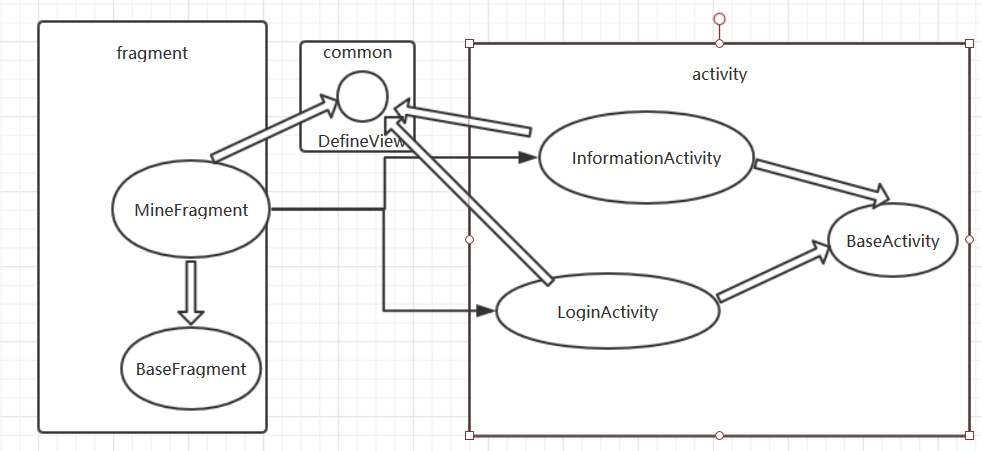
#### 2.3.2.1 HttpUtils

在前端，将所有访问后端的操作都封装入一个Java类中——HttpUtils，当系统需要访问后端时，均通过此类；当系统扩展接口时，只需要将此类和相关类进行更新即可。他是前端与后端沟通的桥梁。它涉及到了很多包里的很多类，如下图所示



#### 2.3.2.2 MineFragment

Activity与Fragment都是android中界面的控制类，我以MineFragment为例介绍各种界面类的切换、接口调用关系。



### 2.3.3数据库接口详情

#### 2.3.3.1获取新闻评论

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 获取数据库中新闻的所有评论 |
| URL | /getNewsAllComment |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| news\_id | true | string | 需要获取所有评论的新闻ID |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success(有评论)/error(没有评论) |
| data | list | 包含该新闻的所有评论，每条评论的信息包含评论作者名(String nick\_name)，作者id(String user\_id)，评论内容(String content)，评论时间(String release\_time) |

#### 2.3.3.2获取好友列表

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 从数据库中获取某个用户的好友列表 |
| URL | /getFriend |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| ID | true | string | ID为需要获取好友列表的用户的ID |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success/error |
| data | list | 包含多个好友的信息，每条好友信息用一个json对象表示，每条信息包含关注的用户的ID(String ID),好友昵称(String nick\_name)字段,按照nick\_name字段升序排列 |

#### 2.3.3.3获取所有用户的动态

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 从数据库中获取所有的动态记录 |
| URL | /getAllTrend |
| 支持格式 | JSON |
| HTTP请求方式 | GET |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| 无 |  |  |  |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success/error |
| data | list | 包含所有动态的具体信息，每条动态信息用一个json对象表示，每条信息包含动态id(String ID)，作者名(String nick\_name)，作者id(String author\_id)，发布时间(String relaese\_time)，动态内容(String content)，关联新闻ID(String news\_id)，新闻title(String title)，新闻url(String url)字段，按照release\_time升序排列 |

#### 2.3.3.4获取用户关注的人的动态

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 从数据库中获取某个用户关注的所有用户发布的动态 |
| URL | /getFriendTrend |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| ID | true | string | 需要获取关注用户发布的动态的用户的ID |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success/error |
| data | list | 包含所有关注用户的动态的具体信息，每条动态信息用一个json对象表示，每条信息包含动态id(String ID)，作者名(String nick\_name)，作者id(String author\_id)，发布时间(String relaese\_time)，动态内容(String content)，关联新闻ID(String news\_id)，新闻title(String title)，新闻url(String url)字段，按照release\_time升序排列 |

#### 2.3.3.5获取自己的动态

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 从数据库中获取某一用户自己发布的所有的动态 |
| URL | /getMyTrend |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| ID | true | string | 需要获取自己动态的用户的ID |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success/error |
| data | list | 包含自己的动态的具体信息，每条动态信息用一个json对象表示，每条信息包含动态id(String ID)，作者名(String nick\_name)，作者id(String author\_id)，发布时间(String relaese\_time)，动态内容(String content)，关联新闻ID(String news\_id)，新闻title(String title)，新闻url(String url)字段，按照release\_time升序排列 |

#### 2.3.3.6获取用户粉丝列表

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 从数据库中获取某一用户的粉丝列表 |
| URL | /getFans |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| ID | true | string | 需要获取粉丝列表的用户的ID |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success/error |
| data | list | 包含多个粉丝的信息，每条粉丝信息用一个json对象表示，每条信息包含粉丝ID(String ID),粉丝昵称(String nick\_name)字段,按照nick\_name字段升序排列 |

#### 2.3.3.6获取用户粉丝列表

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 从数据库中获取某一用户的粉丝列表 |
| URL | /getFans |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| ID | true | string | 需要获取粉丝列表的用户的ID |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success/error |
| data | list | 包含多个粉丝的信息，每条粉丝信息用一个json对象表示，每条信息包含粉丝ID(String ID),粉丝昵称(String nick\_name)字段,按照nick\_name字段升序排列 |

#### 2.3.3.6获取用户粉丝列表

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 从数据库中获取某一用户的粉丝列表 |
| URL | /getFans |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| ID | true | string | 需要获取粉丝列表的用户的ID |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success/error |
| data | list | 包含多个粉丝的信息，每条粉丝信息用一个json对象表示，每条信息包含粉丝ID(String ID),粉丝昵称(String nick\_name)字段,按照nick\_name字段升序排列 |

#### 2.3.3.7 关注用户

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 向数据库中插入一个关注关系元组 |
| URL | /insertInterest |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| user\_id | true | string | 需要添加关注的用户ID |
| friend\_id | true | string | 被关注的用户ID |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success代表添加成功/error代表添加失败 |

#### 2.3.3.8取关好友

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 从数据库中删除一条关注关系元组 |
| URL | /deleteFriend |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| user\_id | true | string | 需要取关的用户ID |
| friend\_id | true | string | 被取关的好友ID |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success代表取关成功/error代表取关失败无好友关系 |

#### 2.3.3.9拉黑粉丝

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 从数据库中删除一条关系元组 |
| URL | /cancelInterest |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| user\_id | true | string | 之前关注发起“拉黑粉丝”动作的用户ID |
| friend\_id | true | string | 被关注的用户ID，发起“拉黑粉丝”动作的用户 |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success代表取消被关注成功/error代表取消被关注失败，无好友关系 |

#### 2.3.3.10判断某个用户是否关注另一个用户

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 从数据库中查询是否存在一个关注关系元组 |
| URL | /judgeFriend |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| user\_id | true | string | 用户1的ID，该用户为主体 |
| friend\_id | true | string | 用户2的ID，该用户为客体 |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success代表用户1关注了用户2/error代表用户1未关注用户2 |
| Is\_friend | int | 0表示用户1未关注用户2，1表示用户1关注用户2 |

#### 2.3.3.11 获取动态信息和相关动态评论

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 从数据库中获取某条动态的详细信息以及它的评论列表 |
| URL | /getTrendInformation |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| news\_id | true | string | 需要获取所有评论的新闻ID |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success(有动态信息)/error(没有动态信息) |
| data | list | 包含该动态的所有信息，动态id(Integer ID)，动态作者id(String author\_id)，动态作者昵称(String nick\_name)，动态内容(String content)，关联新闻id(Integer news\_id) |
| commentList | list | 包含该动态的评论列表，评论列表包括该条动态的所有评论，每条评论的评论作者名(String nick\_name)，评论者id(String user\_id)，评论内容(String content)，评论时间(String release\_time),只显示动态发布者关注的好友发布的评论。没有评论时，为空 |

#### 2.3.3.12 发布动态

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 向数据库中插入一个动态发布元组 |
| URL | /releaseTrend |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| user\_id | true | string | 需要发布动态的user\_id |
| content | true | string | 需要发布的动态内容 |
| news\_id | true | string | 相关的新闻id |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success(发布动态成功)/insert trend error(插入动态表失败)/release trend error(插入发布动态失败) |

#### 2.3.3.13评论动态

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 向数据库中插入评论动态的相关数据 |
| URL | /insertTrendComment |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| user\_id | true | string | 需要发布动态的user\_id |
| comment | true | string | 需要发布的动态内容 |
| trend\_id | true | string | 要评论的动态id |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success(发布动态评论成功)/insert error(动态评论发布失败) |

#### 2.3.3.14获取新闻

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 从数据库中获取每条新闻的详情 |
| URL | /getNews |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| type | true | string | 支持类型：头条/社会/国内/国际/娱乐/体育/军事/科技/财经/时尚。 |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success/error |
| data | list | 包含多个新闻的信息，每条新闻用一个json对象表示，每条新闻包含author,time,title,type,url,picture字段 |

#### 2.3.3.15 处理收藏

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 添加、查询、取消收藏 |
| URL | /collect |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| key | true | string | add:添加收藏，query：查询是否收藏，delete：取消收藏 |
| user | true | string | 用户id |
| news | true | string | 新闻id |

**返回字段：**

返回一个String字符串，对于add和delete结果始终为True，对于query，True表示已收藏，False表示未收藏。对于其他key请求，返回BadRequest。

#### 2.3.3.16获取全部收藏

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 从数据库中获取收藏信息 |
| URL | /getCollect |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| user | true | string | 用户ID |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success/error |
| data | list | 包含多个新闻的信息，每条新闻用一个json对象表示，每条新闻包含author,time,title,type,url,picture字段 |

#### 2.3.3.17登录

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 进行用户登录 |
| URL | /login |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| id | true | string | 用户名 |
| password | true | string | 密码 |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success/error |
| id | string | 用户在数据库中的唯一标识id |

#### 2.3.3.18 搜索新闻

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 完成搜索新闻功能，支持关键词搜索，类型搜索，发布时间搜索 |
| URL | /searchNews |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| type | false | string | 支持类型：头条/社会/国内/国际/娱乐/体育/军事/科技/财经/时尚。 |
| keyword | false | string | 搜索关键词 |
| time | false | string | 新闻发布时间，满足”yyyy-MM-dd hh:mm:ss"的子字符串格式 |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success/error |
| data | list | 包含多个新闻的信息，每条新闻用一个json对象表示，每条新闻包含author,time,title,type,url,picture字段 |

#### 2.3.3.19注册

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 完成注册功能 |
| URL | /register |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| id | true | string | 用户名 |
| passowrd | true | string | 密码 |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success/error， 错误的唯一情况是注册用户名重复 |

#### 2.3.3.20搜索动态

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 完成搜索动态功能，支持作者昵称搜索，关键字搜索，发布时间搜索 |
| URL | /searchTrend |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| friend | false | string | 好友昵称 |
| keyword | false | string | 搜索关键字 |
| time | false | string | 时间 满足”yyyy-MM-dd hh:mm:ss"的子字符串格式 |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success，搜索成功。error，搜索失败，没有搜索结果 |
| data | list | 动态的list列表。每条动态包括id，content，time，comment\_num，news\_id字段 |

#### 2.3.3.21 修改昵称

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 完成用户修改昵称功能 |
| URL | /modifyName |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| user | true | string | 用户id |
| nick\_name | true | string | 新昵称 |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success。当前没有执行规则检查。应当放在前端验证。 |

#### 2.3.3.22修改密码

**基本信息：**

|  |  |
| --- | --- |
| 功能描述 | 完成修改密码功能 |
| URL | /modifyPassword |
| 支持格式 | JSON |
| HTTP请求方式 | POST |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 必选 | 类型 | 说明 |
| user | true | string | 用户id |
| password | true | string | 新密码 |

**返回字段：**

|  |  |  |
| --- | --- | --- |
| 返回字段 | 字段类型 | 说明 |
| status | string | 返回结果状态。success。当前没有执行规则检查。应当放在前端验证。 |

### 2.3.4 系统与外部系统之间的接口

**聚合数据-新闻头条接口：**

该接口提供最新新闻头条，各类社会、国内、国际、体育、娱乐、科技等资讯，更新周期为5-30分钟。通过周期性访问接口获取新闻头条的题目，日期，分类，作者，链接等信息，并写入数据库作为数据库所提供的新闻。

|  |  |
| --- | --- |
| 接口地址 | http://v.juhe.cn/toutiao/index |
| 返回格式 | json |
| 请求方式 | get/post |
| 请求示例 | http://v.juhe.cn/toutiao/index?type=top&key=APPKEY |

## 2.4 界面设计

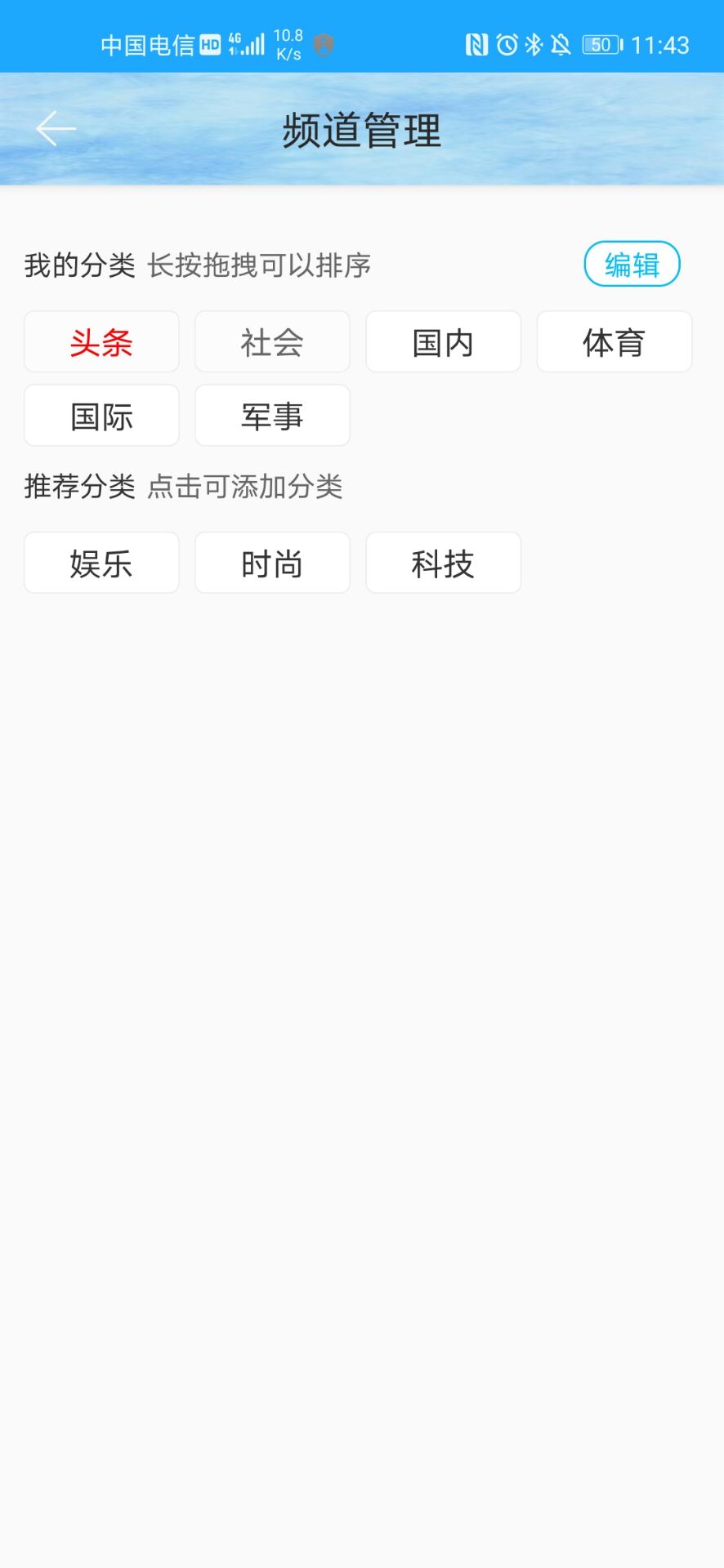
### 2.4.1新闻详情页面

该页面展示新闻标题，作者，发布时间，新闻的具体内容，下拉可浏览全部内容



### 2.4.2频道管理页面

该页面展示了所有频道，点击“编辑”按钮“可以对频道进行增加删除或排序。



### 2.4.3新闻中心页面

该页面展示特定频道内的所有新闻的列表，新闻按照发布时间降序排列。



### 2.4.4评论/收藏/取消收藏新闻页面

该页面展示对新闻的评论/收藏/取消收藏，点击“评论”按钮输入评论，点击“收藏”按钮收藏新闻，点击“取消收藏“按钮取消收藏新闻。





### 2.4.5新闻相关评论页面

该页面显示新闻的所有评论。



### 2.4.6动态搜索

该页面显示了动态的搜索点击“关注“按钮可以按照关注的用户进行搜索，点击”时间“按钮可以按照发布时间进行搜索，点击”搜索“按钮可以按照关键字进行搜索。



### 2.4.7新闻搜索页面

该页面显示了新闻的搜索点击“类型“按钮可以按照新闻类型进行搜索，点击”时间“按钮可以按照发布时间进行搜索，点击”搜索“按钮可以按照关键字进行搜索。



### 2.4.8评论动态

该页面展示了如何让进行动态评论。点击“评论“按钮，输入评论。



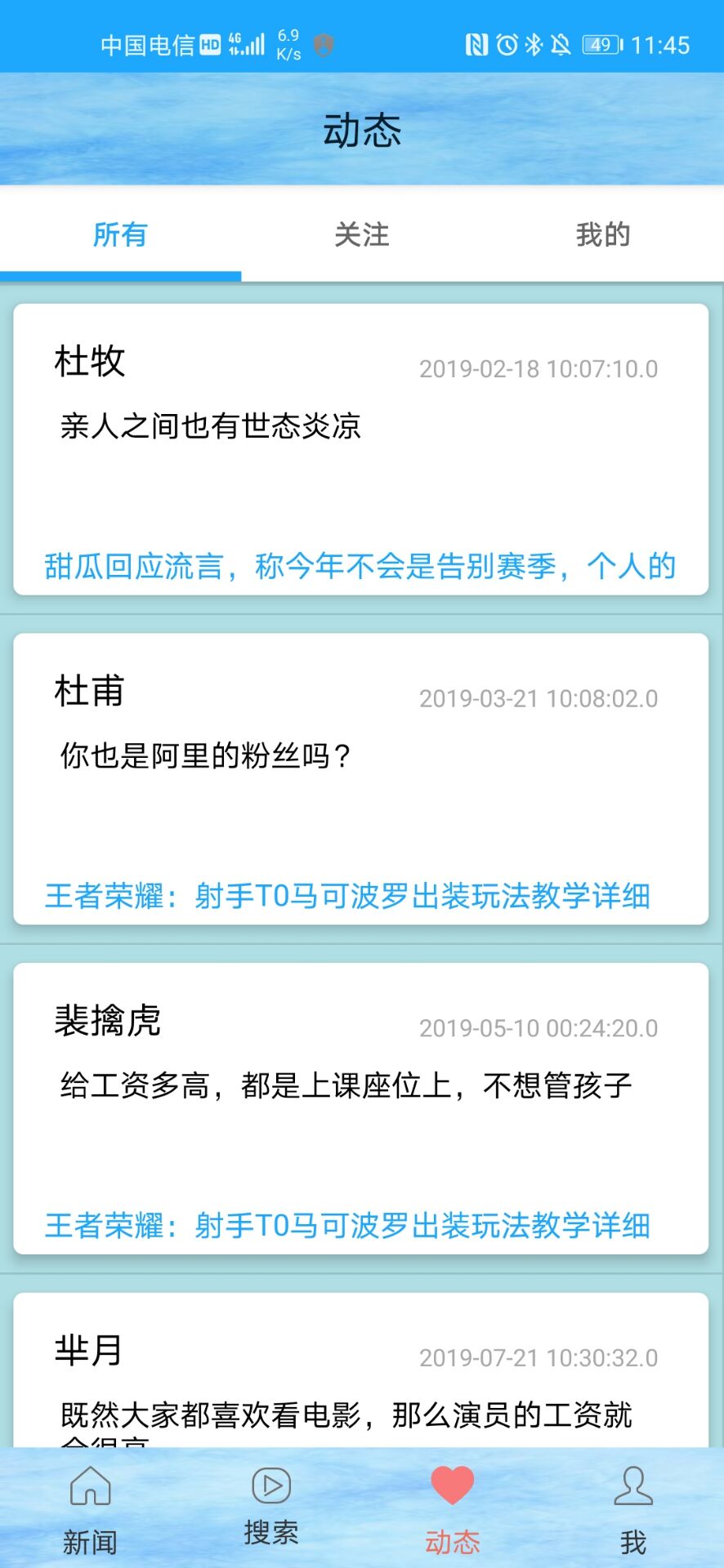
### 2.4.9动态详情页面

该界面展示动态发布时间，作者，内容，相应的新闻标题。



### 2.4.10动态页面

该页面展示动态列表，切换导航栏中的选项可以切管动态列表。



### 2.4.11关注页面

该页面展示用户关注列表。



### 2.4.12个人信息页面

该页面用户可以输入昵称和密码进行修改。



### 2.4.13用户中心页面

该页面展示用户中心。



### 2.4.14登陆注册页面

该页面可以进行登录和注册。



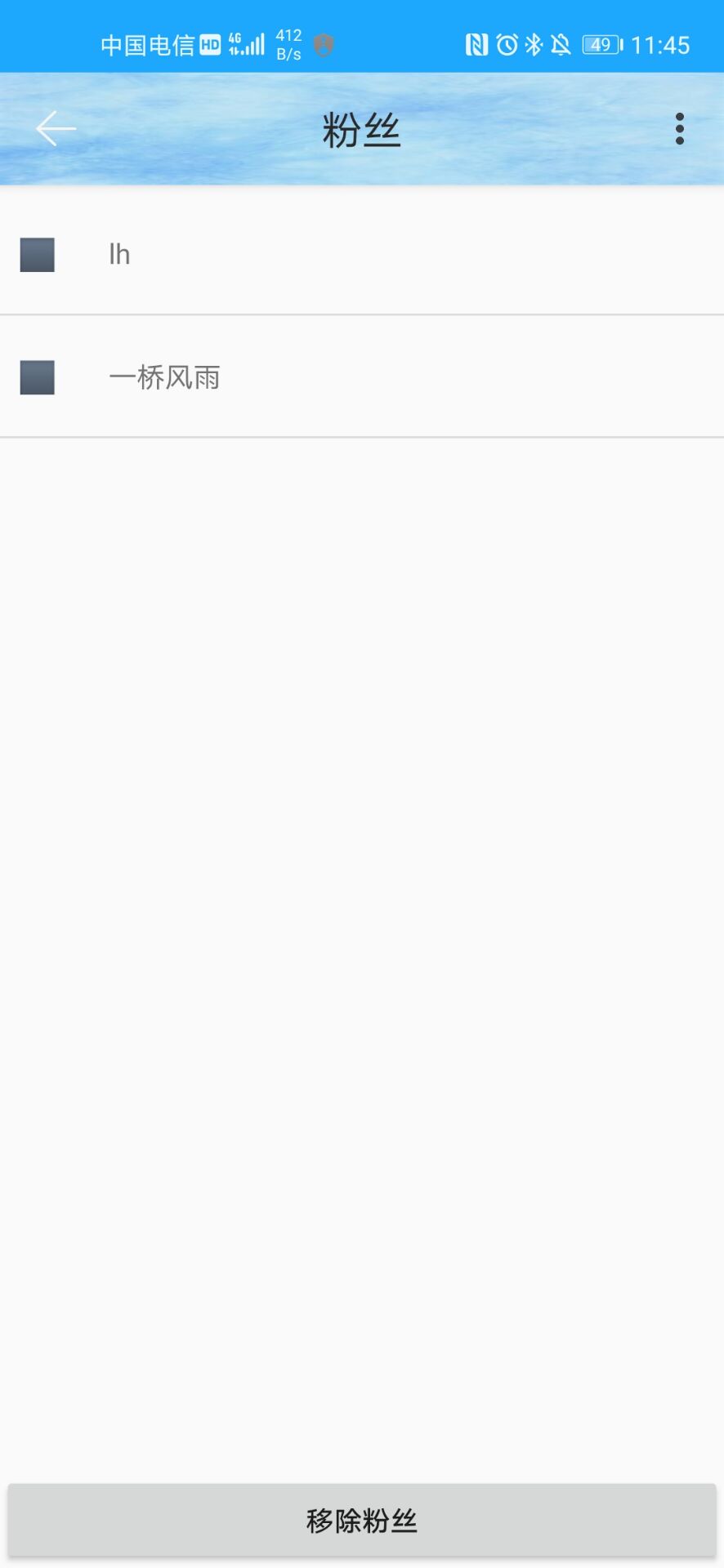
### 2.4.15收藏页面

该页面展示了用户收藏的新闻列表。



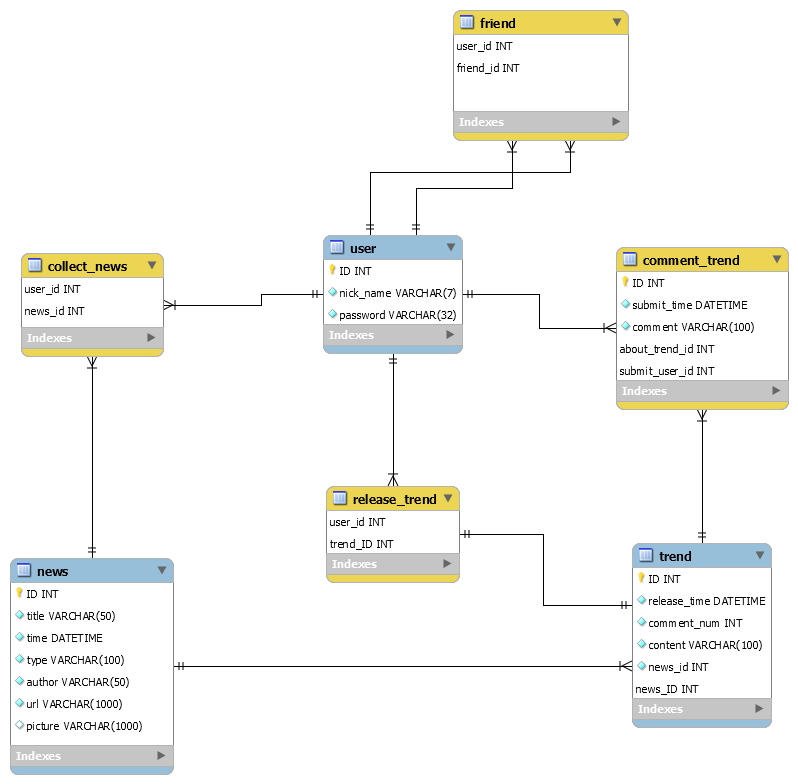
### 2.4.16粉丝页面

该页面显示用户的粉丝列表。

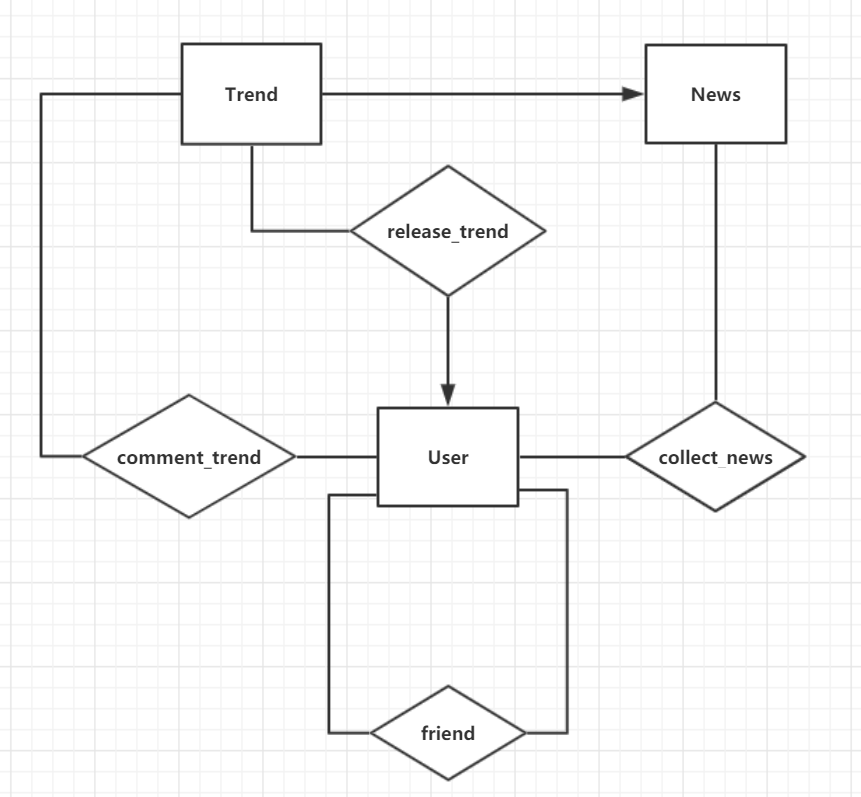


## 2.5 数据库设计

本系统中新闻数据，视频数据，动态数据以及关系数据存储在MySQL数据库中，依据《需求分析规约》中的3.4.2节数据建模，对数据库逻辑进行设计，产生本节中的数据库设计结果。



数据表图



E-R图

### 2.5.1实体关系

一个用户可以关注多个其他用户，一个用户也可以被多个其他用户关注，故user（用户）与user（用户）之间是多对多的关系。因此建立friend关系集，描述用户与用户之间的好友关系。

一个用户可以收藏多条新闻，一个新闻也可以被多个用户收藏，因此user（用户）与news（新闻）之间是多对多的关系。因此建立collect\_news关系集，描述被收藏的新闻与收藏新闻的用户之间的关系。

一个用户可以发布多条动态，但一条动态只能由一个用户发布，故user（用户）与trend（动态）之间是一对多的关系，因此建立release\_trend关系集，描述用户与发布的动态的关系。

一个用户可以评论多条动态，一个动态也可以被多个用户评论，故此时user（用户）与trend（动态）是多对多的关系，因此建立comment\_trend关系集，描述动态的评论与评论动态的用户之间的关系。

### 2.5.2实体属性

**2.5.2.1 user（用户）**

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| ID | int | 用户ID，主键 |
| nick\_name | varchar(7) | 用户昵称 |
| password | varchar(32) | 用户密码 |

**2.5.2.2 collect\_news（用户-新闻关系集：用户收藏新闻）**

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| user\_id | int | 用户ID，主键1 |
| news\_id | int | 被收藏的新闻ID，主键2 |

**2.5.2.3 comment\_trend（用户-动态关系集：用户评论动态）**

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| submit\_user\_id | int | 用户ID，主键1 |
| about\_trend\_id | int | 相关的动态ID，主键2 |
| comment | varchar(100) | 评论内容 |
| submit\_time | datetime | 评论提交时间 |

**2.5.2.4 friend（用户-用户关系集：用户之间的好友关系）**

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| user\_id | int | 用户ID，主键1 |
| friend\_id | int | 好友ID，主键2 |

**2.5.2.5 news（新闻）**

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| ID | int | 新闻ID，主键 |
| title | varchar(500) | 新闻标题 |
| time | datetime | 发布时间 |
| type | varchar(100) | 新闻类型 |
| author | varchar(50) | 新闻作者 |
| url | varchar(1000) | 新闻网址 |
| picture | varchar(1000) | 新闻图片地址 |

**2.5.2.6 release\_trend（用户-动态关系集：用户发布动态）**

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| user\_id | int | 用户ID，主键1 |
| trend\_id | int | 动态ID，主键2 |

**2.5.2.7 trend（动态）**

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| ID | int | 动态ID，主键 |
| release\_time | datatime | 动态发布时间 |
| content | varchar(100) | 动态内容 |
| comment\_num | int | 动态评论数量 |
| news\_ | int | 相关新闻ID |

## 2.6 系统出错处理设置